Actes de CFIP 2008

Colloque Francophone sur l'Ingénierie des Protocoles

du 24 au 28 mars 2008
aux Arcs 1800, France

http://cfip2008.imag.fr

Le mot des présidents du colloque

Le Colloque Francophone sur l'Ingénierie des Protocoles constitue, tous les 18 mois, un lieu de rencontre pour la communauté francophone dans le domaine des réseaux de communication et des méthodes de conception ou d'évaluation associées. CFIP n'a pas cessé de se confirmer et de s'améliorer au fil de ses éditions : Bordeaux (1988), Pau (1991), Montréal (1993), Rennes (1995), Rabat (1996), Liège (1997), Nancy (1999), Toulouse (2000), Montréal (2002), Évry (2003), Bordeaux (2005) et Tozeur (2006). Il témoigne de la vitalité et de la variété de la communauté francophone de recherche sur les bases techniques de la communication.

Organisé aux Arcs dans les Alpes françaises, cette treizième édition du CFIP innove également dans sa méthode de publication. En effet, les actes du colloque sont maintenant publiées en archives ouvertes, dans le cadre du serveur HAL promu au sein du monde scientifique francophone pour la publication de documents scientifiques. Il est significatif que notre colloque qui est au coeur des techniques de l'Internet accompagne cette démarche en se publiant sous cette forme.

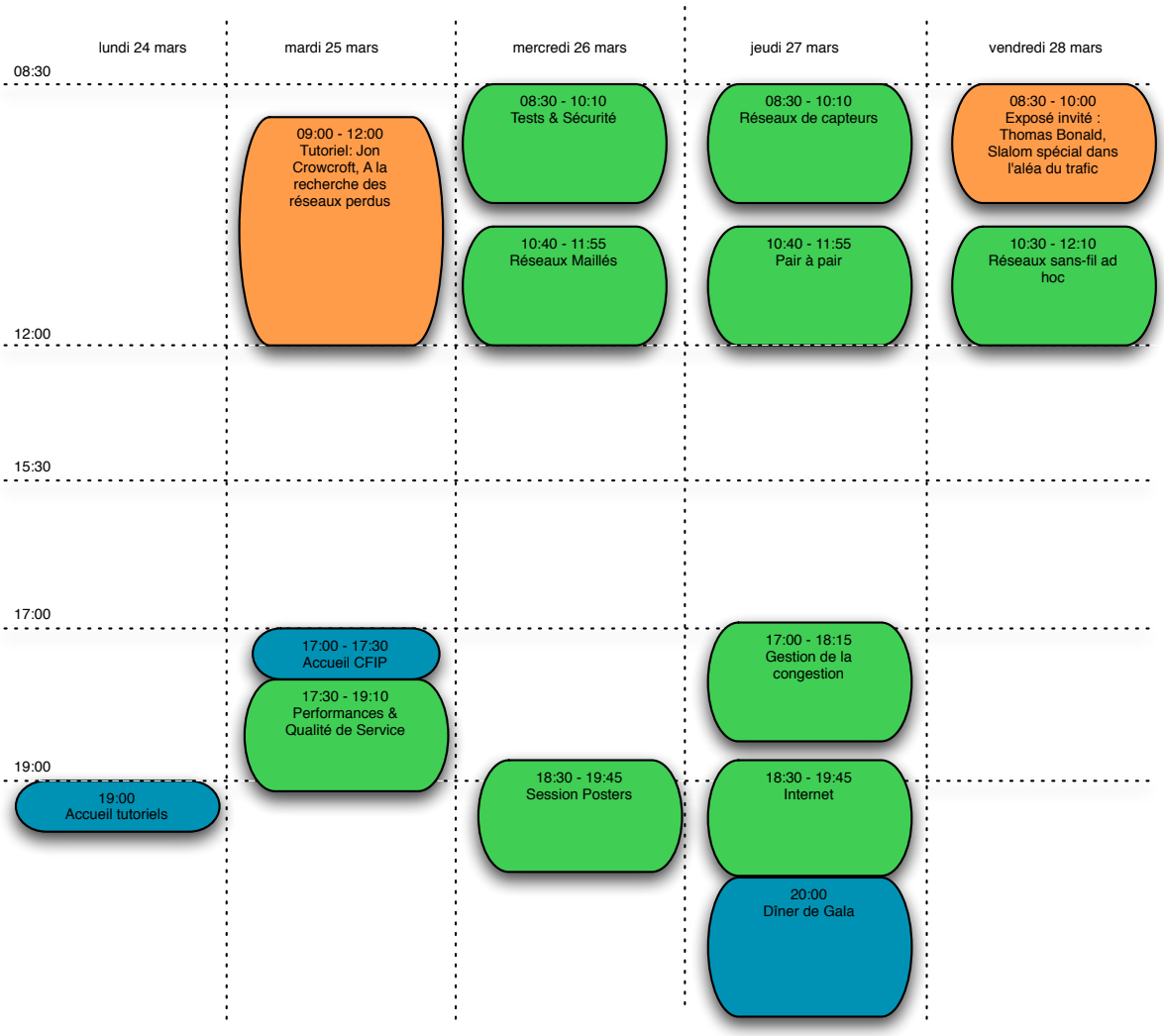
Le comité de programme, constitué de 57 membres issus de 12 pays, a reçu 52 soumissions d'articles de recherche, et en a sélectionné 28 pour présentation au colloque. Par ailleurs, 7 articles courts exposant des travaux en cours seront également présentés sous forme de posters.

Comme toujours, notre colloque est ouvert au-delà de la francophonie à des chercheurs venus d'autres horizons. Ainsi, CFIP 2008 s'ouvre sur un tutoriel présenté par Jon Crowcroft, venu d'Outre-Manche (Cambridge), qui nous invite "À la recherche des réseaux perdus". Les capacités d'inculturation du CFIP se seront également manifestées en fin de colloque, avec l'exposé invité de Thomas Bonald, nous emmenant dans un "Slalom spécial dans l'aléa du trafic".

Nous remercions tous les auteurs qui ont soumis au CFIP les travaux de leur recherche, ainsi que les membres du comité de programme et les relecteurs qui ont permis de présenter cette sélection. Nous remercions également toutes les personnes qui auront contribué à l'organisation de ce CFIP au site des Arcs, en particulier Pascale Poulet du LIG, et les membres de l'équipe Drakkar. Le CFIP 2008 doit en particulier beaucoup à Franck Rousseau et Fabrice Théoleyre qui ont assuré le gros de l'organisation, avec la mise en place du site, la gestion des soumissions et des évaluations, l'édition des actes et des supports, le programme local et l'intendance aux Arcs avec la société partenaire.

Andrzej Duda et **Roland Groz**

Planning de la conférence



Performances & Qualité de Service

1. Thomas Begin et Alexandre Brandwajn. Note sur les temps de service résiduels dans les systèmes type M/G/c
2. Tadeusz Czachórski, Krzysztof Grochla et Ferhan Pekergin. Un modèle d'approximation de diffusion pour la distribution du temps d'acheminement des paquets dans les réseaux de senseurs
3. Baptiste Jacquemin, Pascal Berthou et Thierry Gayraud. QoS et mobilité dans un système DVB-S/RCS
4. Pascale Primet, Jean-Patrick Gelas et Olivier Mornard. Evaluation des performances réseau dans le contexte de la virtualisation

Tests & Sécurité

1. Corinne Sieux, Hervé Aiache, Mattéo Lauriano et Cedric Tavernier. Automatisation de tunnels IPSec imbriqués pour établir des communications anonymes

2. Philippe Owezarski, Silvia Farraposo et Edmundo Monteiro. Détection,Classification et identification d'anomalies de trafic
3. Ana Cavalli, Eliane Martins, Anderson Morais et Bruno Moreira. Une approche de test de robustesse basée sur l'injection de fautes et le monitorage
4. Philippe Owezarski, yann Labit, Pascal Berthou et David Gauchard. LaasNetExp : une plateforme expérimentale pour l'émulation et les tests de réseaux

Réseaux maillés

1. Bezahaf Mehdi Mohamed, Luigi Iannone et Serge Fdida. Gestion avancée de la mobilité dans les réseaux maillés sans fil
2. Christelle Molle, Fabrice Peix et Hervé Rivano. Génération de Colonnes pour le Routage et l'Ordonnancement dans les Réseaux Radio Maillés
3. Hervé Aïache, Vania Conan, Laure Lebrun, Jérémie Leguay, Stéphane Rousseau et Damien Thoumin. Design d'une interface cross-layer et implémentation de métriques complexes pour les réseaux ad-hoc

Réseaux de capteurs

1. François Ingelrest, Guillermo Barrenetxea et Martin Vetterli. SensorScope, un système clef en main de surveillance de l'environnement
2. Antoine Gallais et Jean Carle. Etude et extension des relais de couverture de surface dans les réseaux de capteurs
3. Laurent Toutain, Kevin Perros et Joongsoo Lee. Supprimer le protocole Neighbor Discovery dans les réseaux de capteurs
4. Karel heurtefeux et Fabrice Valois. Localisation collaborative pour réseaux de capteurs

P2P

1. Gilles Tredan et Aline Viana. Systèmes répartis à grande échelle : de l'anarchie à l'auto-structuration
2. Thomas Silverston. Caractérisation du trafic P2P IPTV : une analyse d'échelle
3. Cristian Popi et Olivier Festor. Monitorage et journalisation dynamiques des topologies dans les réseaux hybrides sans fil maillé

Gestion de la congestion

1. Vincent Lucas, Jean-Jacques Pansiot, Dominique Grad et Benoît Hilt. Vers un mécanisme de contrôle de congestion dirigé par les récepteurs pour le multicast
2. Pascal Anelli, Fanilo Harivelo et Emmanuel Lochin. Détection des événements de congestion de TCP
3. Nicolas Van Wambeke, François Armando, Christophe Chassot et Ernesto Exposito. Un modèle de décision pour l'auto-adaptation des protocoles de communication

Internet

1. François Cantin, Mohamed Ali Kaafar, Bamba Gueye, Guy Leduc et Laurent Mathy. Explication et réduction de l'impact des violations d'inégalités triangulaires dans Vivaldi
2. Marc-Olivier Buob, Steve Uhlig et Mickael Meulle. Conception de réseau iBGP
3. Brice Augustin, Farah Layouni, Timur Friedman et Renata Teixeira. Origine et prévention des étoiles dans traceroute

Réseaux sans-fil ad hoc

1. Thierry Val et Adrien Van den Bossche. Développement et analyse multi outils d'un protocole MAC déterministe pour un réseau de capteurs sans fil
2. Rémi Vannier et Isabelle Guerin Lassous. Efficacité d'un modèle simple de partage de bande passante dans les réseaux ad hoc
3. Sofiane Khalfallah, Moez Jerbi, Mohamed Oussama Cherif, Sidi-Mohammed Senouci et Bertrand Ducourthial. Expérimentations des communications inter-véhicules
4. Gilbert Sawma, Ryad Ben-El-Kezadri, Issam Aib et Guy Pujolle. Gestion Autonomique de la Répartition de Charge pour le Contrôle de la Congestion dans les Réseaux Sans Fil

Posters

1. Lucas Nussbaum, Olivier Richard. Prototype de Canal Caché dans le DNS
2. Florent Kaisser, Anthony Busson, Véronique Vèque. Propriétés structurelles des réseaux ad hoc de véhicules
3. Husnain Mansoor Ali, Amina Meraihi Naimi, Anthony Busson, Veronique Veque. Un algorithme de gestion des adjacences basé sur la puissance du signal
4. Fadila Khadar, David Simplot-Ryl. Modèlisation de protocoles MAC pour réseaux de capteurs à l'aide de chaînes de Markov
5. Said Hoceini, Abdelhamid Mellouk, Hayet Hafi, Une approche modulaire probabiliste pour le routage a QdS intégrée
6. Yosra Barouni, Prométhée Spathis, Serge Fdida. Modèle générique pour les réseaux orientés contenu
7. Salma Ktari, Artur Hecker, Houda Labiod. Effet de la mobilité dans un MANET sur système P2P

Performances &
Qualité de Service

Note sur les temps de service résiduels

Dans les systèmes type M/G/c

Thomas Begin[*] - Alexandre Brandwajn[#]

[*] *UPMC Univ Paris 06, UMR (CNRS) LIP6, Paris, France*
thomas.begin@lip6.fr
[#]*University of California Santa Cruz, Baskin School of Engineering, USA*
alexb@soe.ucsc.edu

RÉSUMÉ. *Lorsque l'analyse exacte de certains modèles paraît difficile, on recherche des solutions approchées. C'est le cas de la file M/G/c pour laquelle les paramètres de performances moyens peuvent être calculés de façon approchée en estimant 1) le minimum des temps restants pour terminer les services des clients en cours de service et 2) la probabilité pour un client arrivant dans la file d'avoir de l'attente. Grâce à une série d'exemples numériques, nous faisons apparaître certaines propriétés méconnues de ces deux quantités. En particulier, pour le minimum des résidus, nous montrons qu'il dépend sensiblement des moments d'ordre supérieur du temps de service, au-delà de la moyenne et du coefficient de variation. Plus généralement, une meilleure connaissance de ces deux quantités, objectif principal de cet article, devrait permettre de mieux saisir quels aspects des approximations des files M/G/c mériteraient d'être repensés afin d'améliorer la qualité de leurs résultats.*

ABSTRACT. *Approximations for the mean performance indices for the M/G/c queue rely on the approximate computation of the probability that an arriving request has to wait for service and of the minimum of residual service times if all servers are found busy. Using numerical examples, we investigate properties of these two quantities. In particular, we show that the minimum of residual service times depends on higher order properties, beyond the first two moments, of the service time distribution. Improved knowledge of the properties of the two quantities studied in this paper provides insight into avenues for improving the accuracy of approximations for the M/G/c queue.*

MOTS-CLÉS : *Files multiserveurs, Temps résiduels, Approximation, Moments d'ordre supérieur.*

KEY WORDS: *Multiserver queues, Residual Times, Approximation, Higher-order moments.*

1. Introduction

Les phénomènes de congestion présents dans de nombreux protocoles, systèmes et réseaux informatiques peuvent être représentés par une ou plusieurs files d'attente [HEI 84, ALL 90]. L'élaboration de ces modèles et leur analyse ont permis de mieux concevoir, dimensionner et calibrer les systèmes informatiques actuels.

Pour des files telles que les M/G/1, M/G/c et G/G/c, le calcul de certains paramètres de performance (par exemple, le temps d'attente moyen ou le nombre moyen de clients en attente) devient simple si l'on est capable d'évaluer *les temps de service résiduels* des clients en cours de service. Les temps de service résiduels sont définis comme les temps restants pour terminer les services des clients en cours de service au moment où un nouveau client entre dans la file. En général, dans le cas d'une file monoserveur, on s'intéresse à la valeur moyenne de ces temps résiduels, t_r, appelée le *résidu* de service, et dans le cas des files multiserveurs, on s'intéresse à la valeur moyenne du plus petit des t_r pour l'ensemble des clients en cours de service, c'est-à-dire au *minimum des résidus* $\min(t_r)$.

Le calcul de t_r peut être simple comme c'est le cas pour les files M/M/1 [KLE 75] puisque la propriété sans-mémoire de la distribution exponentielle implique qu'à tout instant le temps moyen qu'il reste au serveur pour terminer son service est égal au temps de service moyen. Dans le cas des files multiserveurs M/M/c [KLE 75], le calcul du minimum des t_r reste simple puisque, ici aussi, la propriété sans-mémoire du temps de service implique qu'à tout instant, le minimum des temps résiduels des k clients en cours de service est identique à celui de k clients qui viendraient de démarrer leur service. Dans le cas d'une file M/G/1, dont le temps de service a une moyenne m et un coefficient de variation cv, le temps de service résiduel d'un client est en moyenne égal à $t_r = \frac{m}{2}\left(1+cv^2\right)$ [KLE 75]. Cette expression très compacte du temps de service résiduel permet d'obtenir des formules simples pour les paramètres de performance moyens d'une file M/G/1.

Toutefois, pour la majorité des files monoserveurs où le processus d'arrivée est non poissonien, ainsi que pour l'ensemble des files multiserveurs dès lors que les temps de service des serveurs ne sont pas distribués exponentiellement, il n'existe aucune expression exacte simple du résidu, ou bien du minimum des résidus lorsqu'il y a plusieurs serveurs. Aussi des chercheurs ont proposé des méthodes pour estimer ces quantités. Ces estimations, dont le domaine de validité peut être restreint, ont permis à leurs auteurs d'obtenir des approximations pour les paramètres de performance moyens de la file considérée [MIY 86, NOZ 78, LAZ 86, EAG 00]. Dans le cas des réseaux fermés, lorsque les hypothèses d'application de l'algorithme MVA (« Mean Value Analysis ») [REI 79, REI 81] ne sont pas rassemblées, il existe des méthodes pour obtenir une solution approchée. Certaines de ces méthodes [HAL 00, LAZ 86, EAG 00] font appel à une estimation des temps résiduels de service qui est calquée sur celle présentée ci-dessus pour la file M/G/1. Pour les files multiserveurs, plusieurs auteurs [MIY 86, HOK 78, HOO 86, MOR 80, NOZ 78] ont proposé des solutions approchées pour le calcul du minimum des résidus en adaptant la formule exacte pour la file M/G/1 au cas multiserveur. Ces études ont permis d'obtenir des approximations pour les

paramètres de performance moyens des files M/G/c ou G/G/c. Toutefois ces approximations ont des domaines d'applicabilité restreints car les estimations du minimum des résidus peuvent dans certains cas s'écarter sensiblement de sa valeur réelle.

Dans cet article, nous nous concentrons particulièrement sur le comportement du minimum des temps de service résiduels t_r dans une file multiserveur à arrivées poissoniennes. Nous obtenons les valeurs de ces quantités grâce à des simulations que nous avons elles-mêmes validées par rapport à une solution semi-numérique exacte permettant le calcul des probabilités des états stationnaires [BRA 07]. Dans la Section 2, nous décrivons la file considérée et nous présentons une technique « classique » pour obtenir une approximation des performances moyennes d'une telle file. Ce type de technique repose sur l'estimation de plusieurs quantités, et notamment sur celle du minimum des résidus. La Section 3 contient les résultats numériques et met en lumière certaines dépendances fortes et méconnues entre le minimum des résidus et les paramètres de la file. En particulier, contrairement à une idée largement admise et reprise en tant qu'hypothèse par les approximations, nous montrons que le minimum des résidus ne dépend pas uniquement de la moyenne et du coefficient de variation du temps de service mais que les moments d'ordre supérieur du temps de service constituent un facteur important dans l'évaluation de cette quantité. Ces dépendances qui semblent méconnues contribuent à la difficulté d'établir correctement le domaine d'applicabilité des approximations pour la file M/G/c. La Section 4 conclut cet article.

2. Système considéré

Tout au long de cet article, nous considérons une file multiserveur à capacité infinie avec discipline de service FIFO. Les arrivées des clients dans cette file sont engendrées par un processus de Poisson de taux λ. La file est composée de c serveurs identiques dont les temps de service sont en moyenne égaux à m. Nous représentons la loi de service de chacun des serveurs par une distribution de Cox à deux étages [COX 61] comme illustré sur la Figure 1. Les taux de service du premier et du second étage de cette distribution sont respectivement donnés par μ_1 et μ_2. $\hat{q}_1$ représente la probabilité pour qu'un serveur poursuive le service d'un client après la conclusion du premier étage. Réciproquement, la probabilité pour qu'un client, une fois le premier étage terminé, quitte la file est donné par $q_1 = 1 - \hat{q}_1$. Le modèle décrit est donc un cas particulier de la file M/G/c pour laquelle le coefficient de variation du temps de service des serveurs est noté cv.

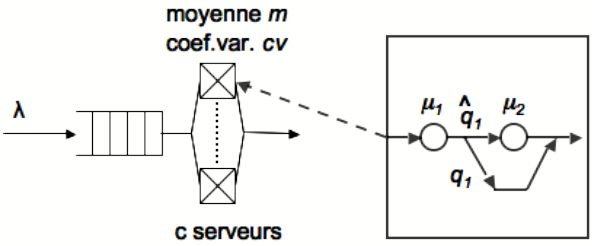


Figure 1. *Modèle considéré*

Nous nous intéressons à l'évaluation des paramètres de performance moyens d'une telle file multiserveur en régime stationnaire. L'utilisation moyenne des serveurs est égale à $\rho = \lambda m/c$ (ρ varie entre 0 et 1). En équilibre, le débit moyen en sortie d'une file sans perte en entrée est connu et toujours égal au débit moyen en entrée, i.e. λ. Le nombre moyen stationnaire de clients dans la file $E(Q)$ peut être exprimé en fonction de ρ et du nombre moyen de clients en attente de libération d'un des serveurs $E(Q_W)$. La loi de Little [KLE 75] permet de remplacer $E(Q_W)$ par le produit du temps qu'un client attend en moyenne avant de démarrer son service, $E(W)$, et du débit moyen λ. Ainsi, on obtient la relation suivante pour $E(Q)$:

$$E(Q) = c\rho + E(W)\lambda$$

Si l'on sait évaluer le temps moyen d'attente avant service d'un client $E(W)$, alors la dérivation des autres paramètres de performance moyens pour une file multiserveur à capacité infinie devient immédiate. Il paraît difficile dans le cas général d'obtenir une formule simple pour $E(W)$. En revanche, certains travaux ont proposé des approximations pour $E(W)$. Soit Π_W la probabilité qu'un client arrivant dans la file trouve tous les serveurs occupés. Rappelons que $\min(t_r)$ représente la valeur moyenne du minimum des résidus des temps de service des clients en cours de service. $E(W)$ peut s'estimer comme :

$$E(W) \approx \Pi_W \min(t_r) + mE(Q_W)/c$$

En appliquant à nouveau la formule de Little sur $E(Q_W)$, on obtient :

$$E(W) \approx \frac{\Pi_W \min(t_r)}{1-\rho} \qquad (1)$$

Ainsi, si l'on connaît les deux quantités Π_W et $\min(t_r)$, l'evaluation de $E(W)$ à partir de (1) ainsi que celle des paramètres de performance moyens d'une file multiserveur à capacité infinie est immédiat. Malheureusement, dans le cas général, il est difficile d'obtenir une expression exacte de ces deux quantités. C'est pourquoi de nombreux travaux en ont proposé des estimations afin d'obtenir des approximations des paramètres de performance moyens pour des files telles que la M/G/c et la G/G/c [MIY 86, HOK 78, HOO 86, MOR 86, NOZ 78].

Hormis l'approximation inhérente à la formule (1), la qualité de ces approximations est directement liée à la précision des estimations de Π_W et $\min(t_r)$. Or, il semblerait que peu de travaux se soient intéressés à évaluer cette précision et à établir les zones d'applicabilité de ces estimations.

A l'aide d'un ensemble d'exemples numériques présentés dans la section suivante, nous mettons en lumière certaines limites de ces approximations, dues notamment à des dépendances fortes et non prises en compte dans l'évaluation approximative de $\min(t_r)$. Pour ce faire, nous utilisons une solution semi-numérique exacte des probabilités des états stationnaires d'une file M/G/c du type considéré [BRA 07], pour obtenir la valeur exacte de la quantité Π_W. Concernant le minimum des residus $\min(t_r)$, nous obtenons sa valeur exacte grâce à des simulations à événements discrets dont nous avons vérifié l'intégrité par rapport à la solution semi-numérique exacte précédemment citée.

3. Résultats numériques

3.1 *Minimum des résidus*

Dans tous les exemples de cette section, les temps de service moyens des serveurs sont exprimés dans une unité temporelle quelconque et égaux à 1 (on rappelle que l'on désigne le moment d'ordre 1 du temps de service par m). Pour chaque étude, nous indiquons le coefficient de variation, le coefficient de dissymétrie (skewness), le coefficient d'aplatissement (kurtosis) du temps de service (quantités associées respectivement aux moments d'ordre 2, 3 et 4), ainsi que la description complète de la distribution type Cox-2 le représentant. Pour chaque figure obtenue par simulation, nous avons représenté à l'aide d'un trait fin les intervalles de confiance estimés à 95%.

3.1.1 *Dépendance vis-à-vis de la distribution du temps de service*

Nous commençons par évaluer les valeurs du minimum des résidus d'une file M/G/c avec 4 serveurs pour une utilisation moyenne de ses serveurs égale à 0.5, i.e. ρ=0.5. La Figure 2 représente l'évolution de $\min(t_r)$ pour des valeurs croissantes du coefficient de variation cv. Les distributions type Cox-2 utilisées dans cet exemple sont décrites dans le Tableau 1. Notons que le taux μ_1 du premier étage de ces distributions est toujours de 1000. Dans la suite de l'article, on désignera cet ensemble de distributions par Dist. I.. Pour les distributions du Tableau 1, les résultats de la Figure 2 semblent indiquer que $\min(t_r)$ augmente lorsque cv augmente, comme c'est également le cas pour la file M/G/1.

cv	μ_1	μ_2	q_1	Skewness	Kurtosis
2	1000	0.400	0.601	3.07	12.77
4	1000	0.118	0.883	6.01	48.28
6	1000	0.054	0.946	9.01	108.30
8	1000	0.031	0.969	12.01	192.43
10	1000	0.020	0.980	15.02	300.63

Tableau 1. *Distributions I*

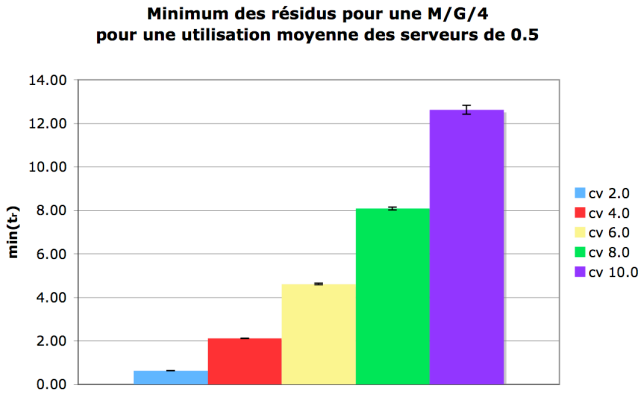


Figure 2. *Influence du coefficient de variation pour Dist. I*

Pour l'étude suivante, nous considérons également une file M/G/4 avec ρ=0.5. Nous évaluons $\min(t_r)$ avec les mêmes valeurs de m et de cv que dans l'exemple précédent, mais les distributions des temps de service ne sont pas les mêmes. La description de ces distributions, que l'on désigne par Dist. II, se trouve dans le Tableau 2. Rappelons que les temps de service considérés dans cet article varient par leur distribution mais que leur moyenne reste toujours égale à 1.0. Contrairement à l'exemple précédent, les $\min(t_r)$ pour cet ensemble de distributions sont petits et restent quasi-constants lorsque le cv du temps de service change comme le montre la Figure 3. La Figure 4 rassemble les résultats des deux exemples précédents sur une seule figure et montre clairement que les moments d'ordre 1 et 2 du temps de service ne suffisent pas pour estimer correctement la valeur de $\min(t_r)$. Par exemple, pour un cv de 4, $\min(t_r)$ peut passer de 0.27 à 2.12 selon que la distribution utilisée pour représenter le temps de service est choisie parmi les Dist. I ou II. On conclut de ces exemples que, à l'inverse d'une file M/G/1, dans les files M/G/c les moments d'ordre supérieur à 2 du temps de service interviennent dans le calcul du minimum des résidus des temps de service en cours que « voit » un client qui rejoint la file.

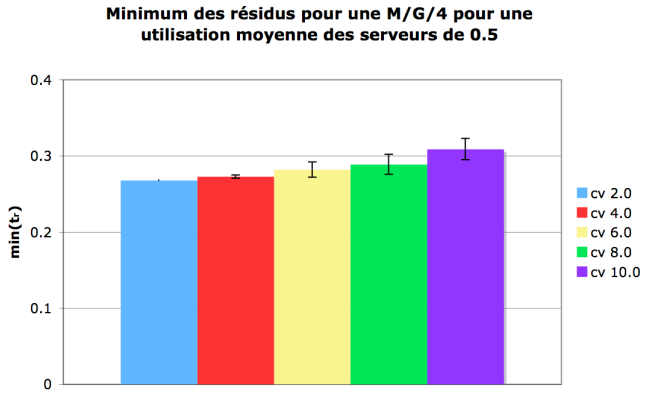


Figure 3. *Influence du coefficient de variation pour Dist. II*

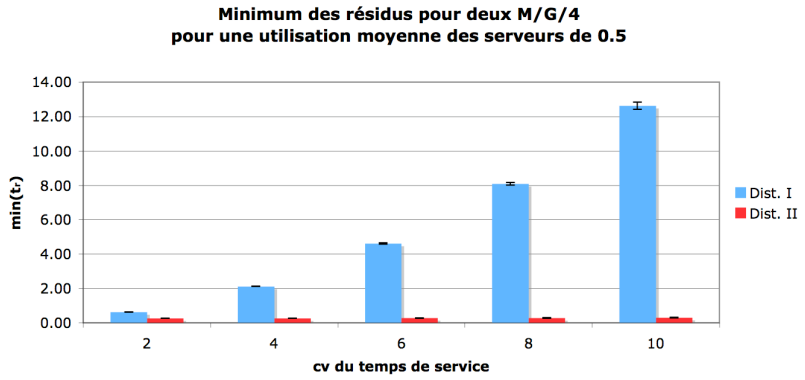


Figure 4. *Influence des moments d'ordre supérieur*

cv	μ_1	μ_2	q_1	Skewness	Kurtosis
2	1.11	0.063	0.9938	19.26	608.91
4	1.11	0.013	0.9987	54.10	4107.30
6	1.11	0.006	0.9994	86.00	10087.28
8	1.11	0.003	0.9997	116.98	18480.19
10	1.11	0.002	0.9998	147.58	29276.89

Tableau 2. *Distributions II*

3.1.2 *Dépendance vis-à-vis de l'utilisation moyenne des serveurs*

A présent, nous vérifions que le minimum des résidus dans une file M/G/c est indépendant de l'intensité de la charge ρ en entrée, comme c'est le cas pour une file monoserveur M/G/1. Pour cela, nous considérons une file M/G/4 à plusieurs niveaux d'utilisation de ses serveurs. La Figure 4 représente $\min(t_r)$ pour les deux distributions du temps de service ayant un cv égal à 4 dans Dist. I et II. Sur cet exemple, et plus généralement sur tous les exemples que nous avons effectués pour des files M/G/c avec des valeurs différentes pour c et cv, nous avons constaté que la valeur de $\min(t_r)$ est indépendante du débit moyen en entrée, ou de manière équivalente, qu'elle ne dépend pas de l'utilisation moyenne des serveurs, c'est-à-dire de ρ.

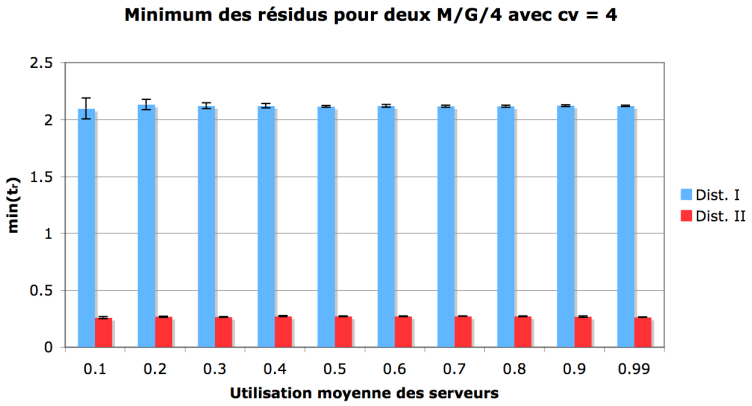


Figure 5. *Minimum des résidus pour plusieurs niveaux de charge en entrée*

3.1.3 *Dépendance vis-à-vis du nombre de serveurs*

A présent, nous nous intéressons à l'évolution de $\min(t_r)$ en fonction du nombre de serveurs dans la file M/G/c. Nous considérons trois distributions différentes du temps de service, mais qui ont toutes la même moyenne et le même coefficient de variation cv. Elles correspondent aux trois distributions ayant cv=4 dans Dist. I, II et III. Notons que dans nos calculs nous avons fixé ρ à 0.5 mais, étant donné les résultats précédents, la valeur de ρ n'importe pas. La Figure 6 représente les résultats obtenus pour un nombre croissant de serveurs allant de 2 à 10. Nous observons assez logiquement que plus le nombre de serveurs est grand, plus le minimum des résidus des services en cours est petit. Plus intéressant, la figure

montre que la vitesse de cette décroissance avec c varie selon la distribution choisie pour le temps de service. Par exemple, le minimum des résidus $\min(t_r)$ est réduit d'un facteur d'environ 2.0 lorsque c passe de 2 à 4 pour la distribution choisie dans Dist. I, tandis que ce facteur passe à 3.9 pour celle choisie dans Dist. III. Plus généralement, nos résultats montrent que cette décroissance n'est pas linéaire, sauf dans certains cas comme ici pour Dist. I. Pour Dist. II et III, la décroissance est plus rapide et elle s'intensifie lorsque c augmente. Ainsi, la décroissance de $\min(t_r)$ avec c dépend des propriétés d'ordre supérieur à 2 de la distribution des temps de service.

cv	μ_1	μ_2	q_1	Skewness	Kurtosis
2	2.5	0.286	0.829	4.64	29.76
4	2.5	0.074	0.956	9.80	129.44
6	2.5	0.033	0.980	14.87	296.05
8	2.5	0.019	0.989	19.90	529.36
10	2.5	0.012	0.993	24.92	829.35

Tableau 3. *Distributions III*

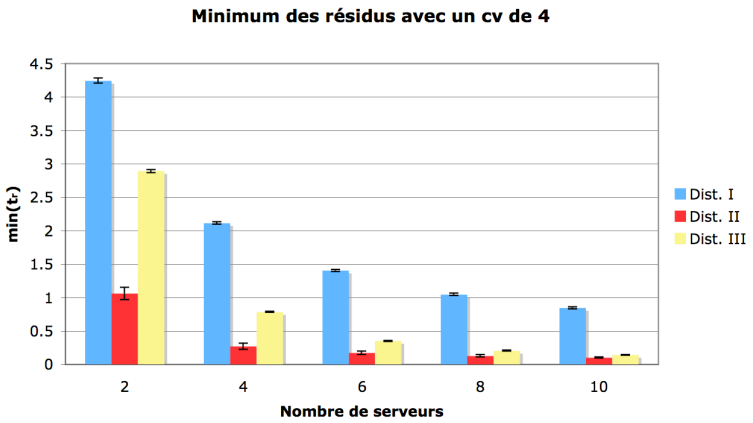


Figure 6. *Influence du nombre de serveurs sur le minimum des résidus*

3.1.4 *Erreur liée à certaines approximations*

Jusqu'ici nous avons montré que le minimum des résidus dans une file M/G/c ne dépend pas uniquement des deux premiers moments m et cv du temps de service. Nos résultats montrent aussi que $\min(t_r)$ a tendance à diminuer lorsque c augmente et que la vitesse de cette diminution varie selon les caractéristiques d'ordre supérieur des distributions du temps de service. Enfin, nos expérimentations corroborent l'idée que, comme on pouvait s'y attendre, $\min(t_r)$ ne dépend pas de l'utilisation moyenne des serveurs d'une M/G/c.

Un certain nombre de solutions approchées pour évaluer les paramètres de performances moyens d'une M/G/c reposent sur des estimations de $\min(t_r)$ et Π_W d'après la formule (1). Pour $\min(t_r)$, les résultats présentés ci-dessus s'opposent à certaines hypothèses faites par ces approximations. D'une part, la plupart de ces

approximations [MIY 86, NOZ 78, LAZ 86, EAG 00] estime la quantité $\min(t_r)$ en prenant en compte uniquement les deux premiers moments du temps de service. Notons que les imprécisions de ces approximations, signalées dans quelques articles [MA 95, BRA 07], s'expliquent peut-être par l'erreur commise initialement dans l'estimation de $\min(t_r)$. D'autre part, certaines approximations estiment la valeur de $\min(t_r)$ pour une file M/G/c [MOR 80, HOK 78] en divisant par c la valeur du temps résiduel trouvée pour une M/G/1 de m et cv identiques. Ceci s'apparente à approcher le minimum de trois variables aléatoires indépendantes par une seule qui serait trois fois plus rapide. Or, si nous avons observé que $\min(t_r)$ a tendance à diminuer lorsque c croît, nous avons également montré que cette décroissance n'est, en général, pas linéaire. On s'attend donc à ce que les approximations qui ne tiennent compte que de m et de cv ainsi que celles qui supposent une relation simple et linéaire entre $\min(t_r)$ et c puissent s'éloigner sensiblement de la valeur exacte de $\min(t_r)$ pour certaines distributions.

La formule la plus simple pour approcher $\min(t_r)$, calquée sur la formule exacte pour estimer t_r dans le cas d'une file M/G/1, et utilisée par certaines approximations [MOR 80, HOK 78] est :

$$\min(t_r) \approx m \frac{(1 + cv^2)}{2c} \qquad (2)$$

Nous étudions maintenant l'erreur liée à cette approximation de $\min(t_r)$ pour toutes les distributions de Dist. I, II et III. Les Figures 7 et 8 représentent respectivement les écarts relatifs obtenus dans le cas d'une file M/G/2 et d'une file M/G/4. Il est intéressant de noter que dans le cas de Dist. I, l'approximation délivre de bons résultats avec des écarts relatifs inférieurs à 1%, tandis que pour Dist. II et III, les écarts sont importants. Par exemple, si on considère la distribution avec un cv de 2 de Dist. II, l'écart atteint environ 100%. Nous observons également que l'erreur semble être plus importante pour les grands cv.

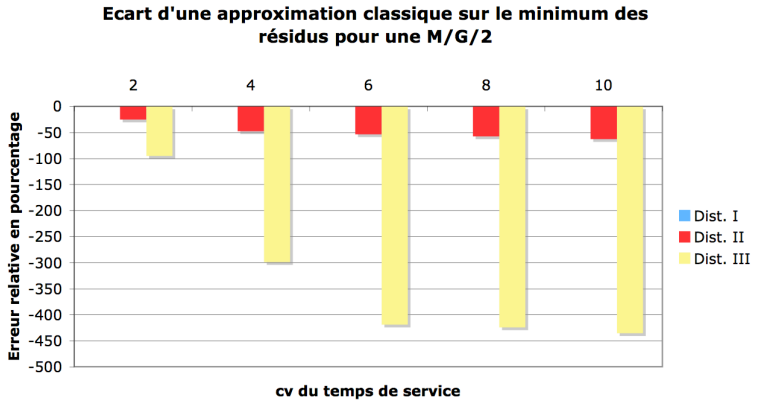


Figure 7. Erreur relative de l'approximation du minimum des résidus

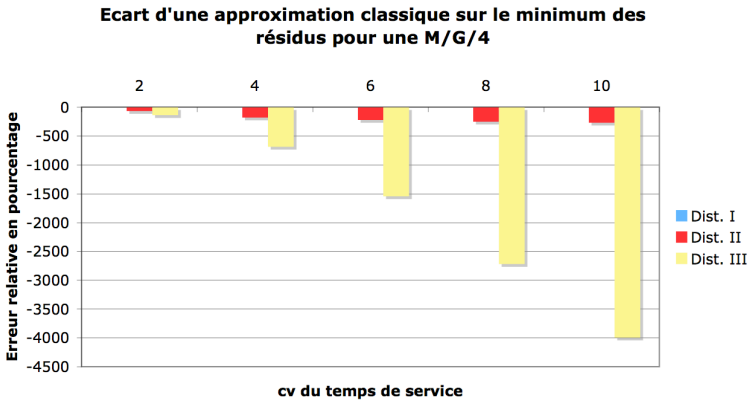


Figure 8. *Erreur relative de l'approximation du minimum des résidus*

3.2 *Probabilité d'attente*

A présent nous nous intéressons à l'autre quantité engagée dans l'évaluation approchée des performances moyennes d'une file M/G/c d'après la formule (1). Il s'agit de Π_w qui désigne la probabilité qu'un client qui arrive dans la file ait de l'attente. Son calcul exact apparaissant compliqué pour une file M/G/c, l'hypothèse de pouvoir remplacer Π_w par la probabilité d'attente dans une M/M/c de même taux est attractive et a été souvent employée [TIJ 81, MIY 86, MOR 80]. Nous savons que cette probabilité, en tant que somme des probabilités des états stationnaires de 0 à c-1, dépend du coefficient de variation et également des moments d'ordre supérieur du temps de service [BRA 07]. Toutefois, nos expériences ont montré que l'écart entre les probabilités d'attente dans les deux files reste modéré (ordinairement compris entre 5 et 10% et pouvant aller jusqu'à 15%) pour un nombre de serveurs allant de 2 à 10, pour des utilisations moyennes des serveurs prises entre 0.1 et 0.99 et des distributions du temps de service choisies parmi Dist. I, II et III. La Figure 9 représente l'erreur liée à l'approximation pour une file avec 4 serveurs et des temps de service choisis dans Dist. III. Nous avons observé que ces écarts ont tendance à s'accroître lorsque cv est grand, et plus intéressant peut-être, nous avons constaté que les erreurs les plus grandes ont lieu pour des utilisations des serveurs de l'ordre de 0.5.

En conclusion de cette section, nous pensons que l'approximation de la valeur de Π_w dans une M/G/c par sa valeur dans une M/M/c convient aux exigences d'une solution approchée qui s'appuie sur la formule (1) pour obtenir les paramètres de performances moyens d'une M/G/c. Concernant l'estimation de $\min(t_r)$, l'omission des moments d'ordre supérieur du temps de service ainsi que la prise en compte parfois trop approximative du nombre de serveurs, peut conduire à des erreurs importantes qui se répercutent dans le calcul des paramètres de performances moyens de la file M/G/c.

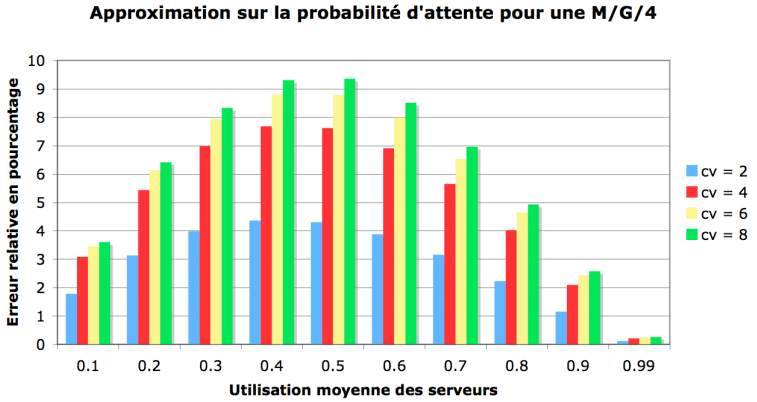


Figure 9. *Erreur relative de l'approximation sur la probabilité* Π_w *pour Dist. III*

4. Conclusions

Dans cet article, nous nous sommes intéressés à deux quantités : le minimum des résidus et la probabilité d'attente d'un client, qui doivent être estimées pour pouvoir appliquer certaines approximations des paramètres de performances moyens d'une file M/G/c [MOR 80, MIY 86, HOO 86, HOK 78, NOZ 78, TIJ 81].

Nous avons montré que le minimum des résidus ne dépend pas uniquement des deux premiers moments du temps de service comme c'est le cas pour une file M/G/1. Des valeurs très différentes du minimum des résidus sont possibles pour des distributions du temps de service ayant la même moyenne et le même coefficient de variation. Toujours pour le minimum des résidus, nous avons montré que son évolution avec le nombre de serveurs dans la file dépend de la distribution du temps de service. Par conséquent, il n'est pas surprenant que les approximations du minimum des résidus qui dérogent à ces propriétés puissent aboutir à des erreurs importantes, y compris pour des valeurs modérées du coefficient de variation de la distribution du temps de service (par exemple, 300% d'erreur relative pour un cv de 4 dans une M/G/2). Par ailleurs, nos résultats ont confirmé que la valeur du minimum des résidus est indépendante de l'utilisation moyenne des serveurs, comme c'est aussi le cas pour la file M/G/1.

Concernant la probabilité d'attente d'un client qui arrive dans une M/G/c, nos résultats permettent de vérifier que la solution courante qui consiste à remplacer cette probabilité par celle obtenue pour une M/M/c de même taux conduit à des écarts modérés, y compris pour des cv importants (environ 5 à 10% d'erreur relative pour des cv allants jusqu'à 8).

Pour conclure, les résultats de nos travaux semblent indiquer que l'amélioration des approximations, pour la file M/G/c, considérées dans cet article impliquent principalement une meilleure estimation du minimum des résidus. Une suite logique à ces travaux consisterait à étendre cette étude à d'autres types de distributions du temps de service et à rechercher quels sont les moments du temps de service dont il faut tenir compte pour obtenir une bonne estimation - voire la valeur exacte - de cette quantité.

5. Bibliographie

[ALL 90] ALLEN A. O., *Probability, statistics and queuing theory with computer science applications,* Vol. 2, San Diego: AcademicPress, 1990.

[BRA 07] BRANDWAJN, A., BEGIN, T., « A Novel Approach to G/C2/c-type queues », Rapport Technique, 2007, University of California Santa Cruz.

[COX 61] COX, D. R., SMITH, W. L. *Queues*. John Wiley, New York, 1961.

[EAG 00] EAGER, D. L., SORIN, D. J., VERNON, M. K. « AMVA techniques for high service time variability », *SIGMETRICS, Perform. Eval. Rev*. Vol. 28, No. 1, 2000, p. 217-228.

[HAL 00] HALACHMI, I., ADAN, I. J. B. F., VAN DER WAL, J., HEESTERBEEK, J. A. P., VAN BEEK, P, « The design of robotic dairy barns using closed queueing networks », *European Journal of Operational Research*, Elsevier, Vol. 124, No. 3, 2000, p. 437-446.

[HEI 84] HEIDELBERGER P., LAVENBERG, S.S., « Computer performance evaluation methodology », *IEEE Trans. Computers*, Vol. 33, 1984, p. 1195-1220.

[HOK 78] HOKSTAD, P., « Approximations for the M/G/m Queue » *Operations Research*, Vol. 26, No. 3. (May - Jun., 1978), p. 510-523.

[HOO 86] VAN HOORN M. H., SEELEN, L. P., « Approximations for the GI/G/c Queue », *Journal of Applied Probability*, Vol. 23, No. 2, 1986, p. 484-494.

[KLE 75] KLEINROCK L., *Queueing Systems Volume 1 : Theory*, Wiley, 1975.

[LAZ 86] LAZOWSKA E. D., ZAHORJAN J., SEVCIK, K. C., « Computer System Performance Evaluation Using Queueing Network Models », *Annual Review of Computer Science*, Vol. 1, 1986.

[MA 95] MA B. N. W., AND MARK J. W., « Approximation of the Mean Queue Length of an M/G/c Queueing System », *Operations Research*, Vol. 43, No. 1, Special Issue on Telecommunications Systems: Modeling, Analysis and Design, 1995, p. 158-165.

[MIY 86] MIYAZAWA, M., « Approximation of the Queue-Length Distribution of an M/GI/s Queue by the Basic Equations », *Journal of Applied Probability*, Vol. 23, No. 2, 1986, p. 443-458.

[MOR 80] MORI M., « Relations between Queue-Size and Waiting-Time Distributions », *Journal of Applied Probability*, Vol. 17, No. 3, 1980, p. 822-830.

[NOZ 78] NOZAKI, S. A., ROSS, S. M., « Approximations in Finite-Capacity Multi-Server Queues with Poisson Arrivals, » *Journal of Applied Probability*, Vol. 15, No. 4, 1978, p. 826-834.

[REI 79] REISER, M., « Mean Value Analysis for Queueing Networks - A New Look at an Old Problem », P*roceedings of the 3rd International Symposium on Modelling and Performance Evaluation of Computer Systems*, 1979, p. 63-77.

[REI 81] REISER, M., « Mean-Value Analysis and Convolution Method for Queue-Dependent Servers in Closed Queueing Networks », *Performance Evaluation,* Vol. 1, 1981, p. 7-18.

[TIJ 81] TIJMS, H. C., VAN HOORN, M. H., FEDERGRUEN, A., « Approximations for the Steady-State Probabilities in the M/G/c Queue », *Advances in Applied Probability*, Vol. 13, No. 1, 1981, p. 186-206.

Un modèle d'approximation de diffusion pour la distribution du temps d'acheminement des paquets dans les réseaux de senseurs

Tadeusz Czachórski[*] — Krzysztof Grochla[*] — Ferhan Pekergin[**]

[*] *IITiS PAN*
44-100 Gliwice, ul. Bałtycka 5, Pologne
{tadek,kgrochla}@iitis.gliwice.pl,
[**] *LIPN, CNRS UMR 7030*
Université Paris-Nord, 93 430 Villetaneuse, France
pekergin@lipn.univ-paris13.fr

RÉSUMÉ. *Nous proposons un modèle qui s'appuie sur l'approximation de diffusion pour l'estimation de la fonction de densité de probabilité du temps d'acheminement des paquets dans un réseaux de senseurs sans fils à sauts multiples. Dans sa forme générale, le modèle tient compte d'une part de la non homogénéité du milieu de propagation et de la répartition des noeuds intermédiaires dans l'espace et d'autre part, des éventuels changements des caractéristiques du réseau dans le temps. La retransmission des paquets perdus est également pris en compte dans le modèle.*

ABSTRACT. *We propose a model based on diffusion aproximation to estimate the probability density function of the distribution of a packet travel time in a multihop wireless sensor network. In its general form, the model assumes that the propagation medium and the distribution of relay nodes may be nonhomogenous in space and that the system characteristics may change over time. The retransmission in case of the packet loss is also considered in the model.*

MOTS-CLÉS : *approximation de diffusion, analyse transitoire, réseaux sans fils, réseaux de senseurs*

KEYWORDS: *diffusion approximation, transient analysis, wireless networks, sensor networks*

1. Introduction

La prédiction du temps d'acheminement des paquets dans les réseaux de senseurs sans fils demeure un problème ouvert. Un réseau de senseurs [AKY 02] consiste en un large nombre de noeuds de structure simple deployés aléatoirement sur une certaine zone et capable de transférer des paquets de voisins en voisins jusqu'au noeud destinataire qui collecte les données qui lui sont transmises par des liaisons à sauts multiples.

Dans la plupart des cas, la topologie d'un tel réseau ne peut pas être connue avec précision d'autant plus qu'elle risque de changer dans le temps à cause du déplacement des noeuds ou de leur défaillance. Le manque de connaissance sur la topologie est à l'origine de nombreux algorithmes de routage spéciaux proposés pour ces réseaux [HEI 00, ROY 99]. La difficulté de mise en place d'un mécanisme d'adressage global implique par ailleurs que les décisions de routage soient prises sans information complète sur le réseau. Nous considérons le modèle proposé dans [GEL 07, GEL 07] : un réseau sans fils composé de noeuds répartis dans une certaine zone mais sans connaissance sur leur existence, l'emplacement et la fiabilité. A chaque noeud les paquets sont transmis vers un autre estimé plus proche de la destination que celui qui émet. Mais cette transmission peut éloigner le paquet de la destination ou peut le transferer à un autre noeud d'une distance égale, nécessitant le même nombre de sauts jusqu'à la destination [ZOR 03]. Il est également possible qu'un paquet ne soit pas retransmis à un noeud intermédiaire à la suite d'une défaillance de ce dernier ou qu'il soit perdu à cause d'une perturbation (bruit) ou d'un autre phénomène transitoire. Dans ce type de situations, le paquet peut être retransmis après un délai de garde soit par la source, soit par un noeud intermédiaire appartenant au chemin emprunté avant la perte et disposant encore une copie du paquet.

2. Modèle de réseau de senseurs

Récemment Gelenbe a proposé [GEL 06, GEL 07] un modèle basé sur l'approximation de diffusion afin d'estimer le temps moyen d'acheminement d'un paquet d'une source à une destination par une transmission aléatoire à sauts multiples. Dans ce modèle le processus de diffusion représente la distance définie par le nombre (minimal) de sauts que le paquet doit effectuer pour atteindre la destination. La complexité de la topologie et les contraintes de la transmission font que la diminution de cette distance au cours d'un saut n'est plus garantie, ainsi, l'évolution de cette distance dans le temps peut être considérée comme un processus aléatoire. Cette remarque rend possible la caractérisation de cette distance par un processus de diffusion. L'approximation de diffusion est une méthode classique en théorie des files d'attente, utilisée pour représenter la longueur d'une file [GEL 75] ou le temps d'attente [GEL 79] dans le cas des distributions générales et indépendantes des temps d'interarrivées et de service. Le processus de diffusion est un processus stochastique à valeurs continues mais il peut servir d'une représentation approchée pour certains processus à valeurs discrètes

comme le nombre d'entités qui attendent dans une file. Dans le cas du modèle en question, il représente le nombre (minimal) de sauts restant à faire jusqu'à la destination.

Soit $N(t)$ le nombre de sauts restant à faire à l'instant t. On définit le processus de diffusion $X(t)$ tel que la fonction de densité de probabilité associée, $f(x,t;x_0)$, approche la distribution de probabilité $p(n,t;n_0)$ du processus $N(t)$, $N(0) = n_0$, $f(n,t;n_0) \approx p(n,t;n_0)$. La fonction de densité

$$f(x,t;x_0)\,dx = P[x \leq X(t) < x + dx \mid X(0) = x_0]$$

est définie par l'équation de diffusion

$$\frac{\partial f(x,t;x_0)}{\partial t} = \frac{\alpha}{2}\frac{\partial^2 f(x,t;x_0)}{\partial x^2} - \beta\frac{\partial f(x,t;x_0)}{\partial x}\ , \qquad [1]$$

où les paramètres β et α sont respectivement la moyenne et la variance infinitésimales du processus de diffusion. Afin d'établir la correspondance avec le processus $N(t)$, ces paramètres doivent être choisis comme

$$\beta = \lim_{\Delta t \to 0} \frac{E[N(t+\Delta t) - N(t)]}{\Delta t} \qquad [2]$$

$$\alpha = \lim_{\Delta t \to 0} \frac{E[(N(t+\Delta t) - N(t))^2] - (E[N(t+\Delta t) - N(t)])^2}{\Delta t} \qquad [3]$$

En général, les paramètres peuvent dépendre du temps et de la valeur du processus, $\beta = \beta(x,t)$ et $\alpha = \alpha(x,t)$, puisque le medium de propagation et la répartition des noeuds relais dans l'espace peuvent être non uniformes et les caractéristiques du système peuvent changer avec le temps. Le modèle proposé tient compte de ces aspects.

Gelenbe en [GEL 07] construit un processus ergodique répétitif évoluant du point initial vers l'origine et considère les propriétés stationnaires de ce processus. Dans ce travail nous obtenons la distribution du temps de transmission (au lieu de sa moyenne) grâce à la solution transitoire de l'équation de diffusion en n'utilisant qu'un seul processus. Rappelons que le processus commence à $x_0 = N$ et se termine à la barrière absorbante située à $x = 0$. La position x du processus correspond à la distance courante mesurée en sauts, entre le paquet et sa destination. On indique les probabilités de reduire le chemin avec π_{-1}, de garder la distance avec π_0, et d'augmenter la distance d'une unité avec π_{+1}. Dans le cas de temps de transmission constant entre deux noeuds, les expressions pour β et α s'obtiennent selon équations (2), (3) à partir de π_{-1}, π_0 et π_{+1} comme :

$$\beta = \frac{\pi_{-1}\times(-1) + \pi_0\times(0) + \pi_{+1}\times(+1)}{\text{1 unité de temps}} = \pi_{+1} - \pi_{-1},$$

$$\alpha = \frac{\pi_{-1}\times(-1)^2 + \pi_0\times 0^2 + \pi_{+1}(+1)^2}{\text{1 unité de temps}} - \beta^2 = \pi_{+1} + \pi_{-1} - \beta^2,$$

et vice-versa :

$$\pi_0 = 1 - \alpha - \beta^2, \qquad \pi_{-1} = (\alpha - \beta + \beta^2)/2, \qquad \pi_{+1} = (\alpha + \beta + \beta^2)/2,$$

à condition que α et β satisfassent :

$$0 \leq \alpha \leq 1, \qquad -1 \leq \beta \leq 1, \qquad |\beta| \leq \alpha + \beta^2 \leq 1.$$

On remarque que pour un temps de transmission constant α et β ne peuvent pas avoir une valeur absolue dépassant 1 et α naturellement, ne peut jamais être negatif.

3. Modèle sans délai de garde et sans pertes

Dans ce cas simple nous considérons l'équation de diffusion (1) avec des coefficients constants et une barrière absorbante placée à $x = 0$. La barrière se traduit dans le modèle par la condition limite $\lim_{x\to 0} f(x,t;x_0) = 0$. Le processus commence à x_0, $X(0) = x_0$, et se termine quand il atteint la barrière. Le processus de diffusion est défini dans l'intervalle $(0,\infty)$. Exprimons par $\phi(x,t;x_0)$ la solution de l'équation de diffusion pour ce cas ; elle s'obtient par la méthode de mirroirs [COX 65],

$$\phi(x,t;x_0) = \frac{1}{\sqrt{2\Pi\alpha t}}\left[e^{-\frac{(x_0-x-|\beta|t)^2}{2\alpha t}} - e^{\frac{2\beta x_0}{\alpha}} e^{-\frac{(2x_0-|\beta|t)^2}{2\alpha t}}\right].$$

La fonction $\phi(x,t;x_0)$ est illustrée à la figure 1. Elle permet de déterminer la densité $\gamma_{x_0,0}(t)$ de temps du premier passage de $x = x_0$ à $x = 0$ et d'exprimer par l'aide de ceci, la densité du temps de transmission d'un paquet d'une distance de $n = x_0$ sauts à la destination :

$$\gamma_{x_0,0}(t) = \lim_{x\to 0}[\frac{\alpha}{2}\frac{\partial}{\partial x}\phi(x,t;x_0) - \beta\phi(x,t;x_0)] = \frac{x_0}{\sqrt{2\Pi\alpha t^3}} e^{-\frac{(x_0-|\beta|t)^2}{2\alpha t}}.$$

Quelques courbes type pour $\gamma_{x_0,0}(t)$ sont représentées à la figure 2.

4. Introduction du délai de garde au modèle

Soit T le délai à l'expiration duquel un paquet est considéré perdu et retransmis par la source. Connaissant la densité $\gamma_{x_0,0}(t)$ du temps de premier passage de x_0 à 0, on peut déterminer la probabilité p_T qu'un paquet ne soit toujours pas arrivé à sa destination au bout d'une durée T : $p_T = \int_T^\infty \gamma_{x_0,0}(t)dt$, figure 3.

Ainsi, quand le délai de garde expire à l'instant $t = T$, on replace la masse de probabilité p_T à x_0 et on recommence le processus de diffusion en introduisant éventuellement un délai additionnel avant la reprise.

5. Modélisation d'un medium nonhomogène et sujet aux pertes

Afin de reflèter la possibilité de changement des conditions de transmission d'un saut à l'autre, le support du processus de diffusion est divisé en intervalles unitaires,

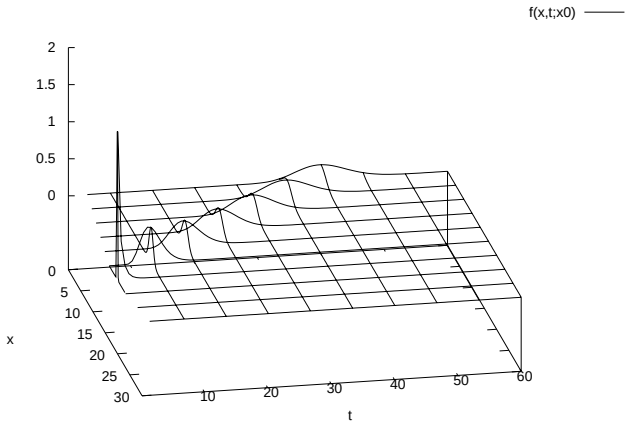


Figure 1. *Fonction de densité $\phi(x, t; x_0)$ du processus de diffusion avec une barrière absorbante ;* $x_0 = 20$, $\alpha = 0.1$, $\beta = -0.5$.

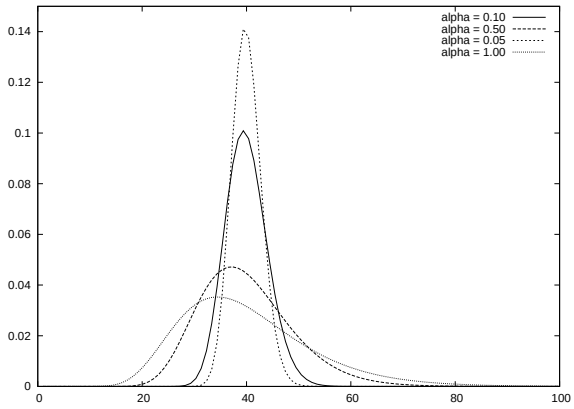


Figure 2. *Distribution du temps de premier passage de x_0 à 0, $\gamma_{x_0,0}(t)$.* $x_0 = 20$, $\beta = -0.5$, $\alpha = 0.05, 0.1, 0.5$ *et* 1.0.

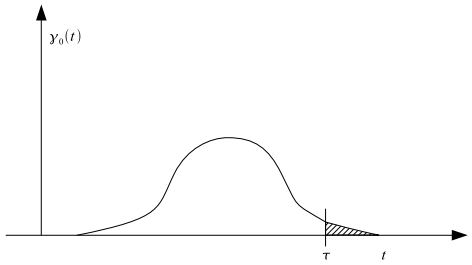


Figure 3. *Probabilité $p_T = \int_T^\infty \gamma_{x_0,0}(t)dt$ qu'un paquet ne soit toujours pas arrivé à la destination après la durée T.*

chacun correspondant à un bond. Les sous-intervalles sont séparés par des barrières fictives dans le but d'équilibrer les flux de densité de probabilité qui les traversent. L'intervalle associé à la totalité du processus est limité à une valeur D correspondant à la taille du réseau, $x \in [0, D]$. Le point de départ x_0 est choisi comme un point correspondant à un entier dans cet intervalle. En général $\beta < 0$, puisque normalement le paquet doit s'approcher de la destination à chaque saut, par conséquent la probabilité que le processus accède à l'extrémité droite du support est très faible. Néanmoins, pour éviter toute perte de probabilité, le processus est immédiatement replacé au point situé à la distance $x = D - \varepsilon$ s'il atteint l'extrémité droite du support, et il continue son évolution par cette position.

Un sous-intervalle i, $x \in [i-1, i]$, représente la transmission du paquet par un noeud situé à une distance i de la destination. Nous admettons que les paramètres β_i, α_i peuvent avoir des valeurs différentes pour chaque sous-intervalle, ainsi que l_i exprimant la probabilité de perte au cours du saut correspondant à cet intervalle.

Quand le processus atteint une barrière $i \neq 0, D$, il est instantanément absorbé. Mais, avec la probabilité $(1 - l_i)$ il réapparait immédiatement à l'autre côté de la barrière (transmission sans problème) et avec la probabilité l_i restante, il retourne à la barrière $x = i + 1$ ou $x = i - 1$, correspondant à la reprise de la transmission du paquet par le noeud qu'il a visité en dernier.

Soit $\gamma_i^L(t)$ le flux entrant dans la barrière $x = i$ par le côté gauche et $\gamma_i^R(t)$ le flux entrant à cette barrière par la droite. Ces flux donnent naissance à des processus de diffusion aux deux côtés de la barrière, $\gamma_i^R(t)$ réapparaît à $x = i - \varepsilon$ et $\gamma_i^L(t)$ à $x = i + \varepsilon$, mais les taux de ces flux traversant la barrière sont réduits proportionnellement avec la probablilté qu'une perte survienne pendant le saut précédent. Ainsi, pour l'intervalle i on définie les densités de probabilité $g_{i-1+\varepsilon}(t)$ et $g_{i-\varepsilon}(t)$ que le processus de diffusion recommence à l'instant t respectivement au point $x = i - 1 + \varepsilon$ et $x = i - \varepsilon$:

$$g_{i-1+\varepsilon}(t) \quad = \quad (1 - l_i)\gamma_i^L(t) + l_{i-1}\gamma_{i-1}^R(t)$$

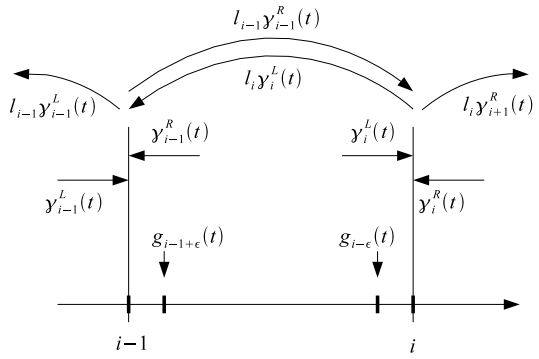


Figure 4. *Le transfert de masse de probabilité dans le cas de paramètres non homogènes de diffusion et des pertes d'un taux l_i au saut i pour $i = 2, \ldots, D-1$*

$$g_{i-\varepsilon}(t) \quad = \quad (1-l_{i-1})\gamma_{i-1}^R(t) + l_i\gamma_i^L(t)$$

Si nous supposons que le problème survenu pendant la transmission peut être reglé par la reprise de l'émission par le noeud voisin, le flux $\gamma_i^L(t)l_i$ est envoyé à $x = i-1+\varepsilon$ et le flux $\gamma_i^R(t)l_i$ à $x = i+1-\varepsilon$. Le mouvement de la masse de probabilité pour l'intervalle i représentant un saut à une distance i est illustré à la figure 4.

Si les paquets sont retransmis après un certain délai aléatoire distribué selon la fonction de densité $l(t)$, (pour un délai constant et égal à r, $l(t) = \delta(t-r)$), l'équation précédente s'écrit comme :

$$\begin{aligned} g_{i-1+\varepsilon}(t) &= (1-l_i)\gamma_i^L(t) + l_{i-1}\gamma_{i-1}^R(t) * l(t) \\ g_{i-\varepsilon}(t) &= (1-l_{i-1})\gamma_{i-1}^R(t) + l_i\gamma_i^L(t) * l(t) \end{aligned}$$

où, $*$ désigne l'opération de convolution.

Dans chaque sous-intervalle on a un processus de diffusion avec deux barrières absorbantes, situées pour le sous-intervalle i à $x = i-1$ et $x = i$, et deux points de génération à $x = i-1+\varepsilon$ et $x = i-\varepsilon$ où le processus recommence avec les taux respectifs de $g_{i-1+\varepsilon}(t)$ et $g_{i-\varepsilon}(t)$. La densité du processus de diffusion commençant à x_0 dans l'intervalle $(0, N)$ qui a deux barrières absorbantes aux extrémités $x = 0$ et $x = N$ est de la forme suivante, [COX 65]

$$\phi(x,t;x_0) = \begin{cases} \delta(x-x_0), t = 0 \\ \dfrac{1}{\sqrt{2\Pi\alpha t}} \displaystyle\sum_{n=-\infty}^{\infty} \Big\{ \exp\left[\frac{\beta x_n'}{\alpha} - \frac{(x-x_0-x_n'-\beta t)^2}{2\alpha t}\right] \\ \quad - \exp\left[\frac{\beta x_n''}{\alpha} - \frac{(x-x_0-x_n''-\beta t)^2}{2\alpha t}\right] \Big\}, t > 0\,, \end{cases}$$

où $x_n' = 2nN$, $x_n'' = -2x_0 - x_n'$.

La densité $f_i(x,t;\psi)$ peut être exprimée comme une superposition des fonctions $\phi_i(x,t;x_0)$ dans l'intervalle $(i-1,i)$

$$\begin{aligned} f_i(x,t;\psi_i) &= \phi(x,t;\psi_i) + \int_0^t g_{i-1+\varepsilon}(\tau)\phi(x,t-\tau;i-1+\varepsilon)d\tau \\ &\quad + \int_0^t g_{i-\varepsilon}(\tau)\phi(x,t-\tau;i-\varepsilon)d\tau \ . \end{aligned}$$

où la fonction ψ_i représente la condition initiale. Pour l'intervalle i, les flux $\gamma_{i-1}^L(t)$ et $\gamma_i^R(t)$ s'obtiennent comme

$$\begin{aligned} \gamma_{i-1}^R(t) &= \lim_{x\to(i-1)}[\frac{\alpha_i}{2}\frac{\partial f_i(x,t;\psi_i)}{\partial x} - \beta_i f_i(x,t;\psi_i)] \\ \gamma_i^L(t) &= -\lim_{x\to(i)}[\frac{\alpha_i}{2}\frac{\partial f_i(x,t;\psi_i)}{\partial x} - \beta_i f_i(x,t;\psi_i)] \end{aligned}$$

La solution du système correspondant à l'ensemble des équations ci-dessus est plus facile à obtenir en utilisant leurs transformées de Laplace. Par exemple, en posant $A(s) = \sqrt{\beta^2 + 2\alpha s}$, la transformée de Laplace de la fonction $\phi(x,t;x_0)$ est

$$\begin{aligned} \bar{\phi}(x,s;x_0) &= \frac{\exp[\frac{\beta(x-x_0)}{\alpha}]}{A(s)} \sum_{n=-\infty}^{\infty} \left\{ \exp\left[-\frac{|x-x_0-x_n'|}{\alpha} A(s) \right] \right. \\ &\quad \left. - \exp\left[-\frac{|x-x_0-x_n''|}{\alpha} A(s) \right] \right\} . \end{aligned}$$

Dans ce cas, toutes les convolutions deviennent des produits des transformées et ainsi la solution $f_i(x,t;\psi_i)$ est obtenue par l'inversion de sa transformée de Laplace $\bar{f}_i(x,s;\psi_i)$.

Pour l'inversion nous utilisons un algorithme simple, l'algorithme de Stehfest, qui donne la valeur d'une fonction $f(t)$ à partir de sa transformée $\bar{f}(s)$ pour chaque valeur fixe de l'argument t, par l'expression :

$$f(t) = \frac{\ln 2}{2} \sum_{i=1}^{N} V_i \ \bar{f}\left(\frac{\ln 2}{t} i \right) \ ,$$

où

$$V_i = (-1)^{H/2+i} \times \sum_{k=\lfloor \frac{i+1}{2} \rfloor}^{\min(i,H/2)} \frac{k^{H/2+1}(2k)!}{(H/2-k)!k!(k-1)!(i-k)!(2k-i)!}.$$

Ici H est un entier pair dépendant de la précision de l'ordinateur ; nous avons pris $H = 14$ dans nos calculs.

Nous montrons dans les figures 5 et 6 la probabilité $p(0,t) = \int_0^t \gamma_{x_0,0}(\tau)\, d\,\tau$ que le processus soit fini (donc la transmision terminée) avant temps t pour les différents

valeurs de paramètres α et β. A la Fig. 5, le paramètre α est maintenu constant et β est varié de -0.2 à -0.8. Comme il est attendu, avec β de plus en plus négatif (routage plus directif vers la destination), le temps d'acheminement diminue considérablement. Il est à noter que, grâce à la supposition de *réseau fermé d'une taille finie*, la distribution du temps d'acheminement existe non seulement pour $\beta = 0$, mais aussi pour les valeurs positive de ce paramètre. En revanche, la courbe pour $\beta = -0.2$ montre clairement que pour $\beta \geq 0$, le temps d'acheminement risque de prendre des valeurs inacceptables. La figure 6 montre que pour un même de diminution de la distance vers la destination, une variabilité plus importante favorise le raccourcissement du temps d'acheminement. En gardant la même vitesse moyenne $\beta = \pi_{+1} - \pi_{-1}$, plus la probabilité π_{-1} de choisir une bonne direction est élevée, plus le temps d'acheminement raccourcit en moyenne malgré l'augmentation de la probabilité des mauvais choix. Ce résultat justifie l'utilisation des routages aléatoires dans les réseaux de senseurs.

Figure 7 présente la densité $f(x, t; x_0)$ de la distance restante à parcourir au paquet aux instants $t = 10, 20$ et 30 pour un réseau caractérisé avec les paramètres $\beta = -0.4$ et $\alpha = 0.54$. Le paquet etait initialement à une distance de $x_0 = 10$ de la destination. Figure 8 nous montre la densité $\gamma_{x_0,0}(t)$ du temps du premier passage entre x_0 et $x = 0$, donc l'approximation du temps de transmission entre le point du départ et la destination. Les valeurs choisies dans cet exemple sont :

$$\begin{aligned}
\beta = -0,4, \alpha = 0.54 \quad &\longrightarrow \quad (\pi_{-1}, \pi_0, \pi_{+1}) = (0.55, 0.3, 0, 15), \\
\beta = -0.2, \alpha = 0.54 \quad &\longrightarrow \quad (\pi_{-1}, \pi_0, \pi_{+1}) = (0.4, 0.4, 0, 2), \\
\beta = -0.3, \alpha = 0.81 \quad &\longrightarrow \quad (\pi_{-1}, \pi_0, \pi_{+1}) = (0.6, 0.1, 0, 3).
\end{aligned}$$

Figure 9 montre l'influence du taux de pertes l ($l = 0.05, 0.1, 0.2, 0.3$) sur les densités du temps d'acheminement de paquet pour le réseau de paramètres $\beta = -0.4$ et $\alpha = 0.54$. Cette sensibilité du temps d'acheminement aux pertes est due à l'assimilation des cas de pertes et de reprise qui les succèdent, aux transmissions vers des noeuds de même distance de la destination que l'émetteur. Par conséquent, conformément à l'effet de la variabilié sur le temps d'acheminement (figure 6), la probabilité de pertes qui s'ajoute à la probabilité π_0, augmente considérablement le temps total de transmission.

6. Conclusion

Grâce à l'analyse des états transitoires, le modèle proposé dans ce travail permet de prendre en compte plus de paramètres (paramétres dépendants du temps et de la position des noeuds, taux de pertes différents pour les sauts) concernant le temps d'acheminement dans les réseaux de senseurs, que les modèles existants, y compris ceux qui s'appuient sur l'approximation de diffusion. Les résultats que l'on obtient sont également plus détaillés : le temps d'acheminement est caractérisé par sa fonction de densité au lieu de sa valeur moyenne.

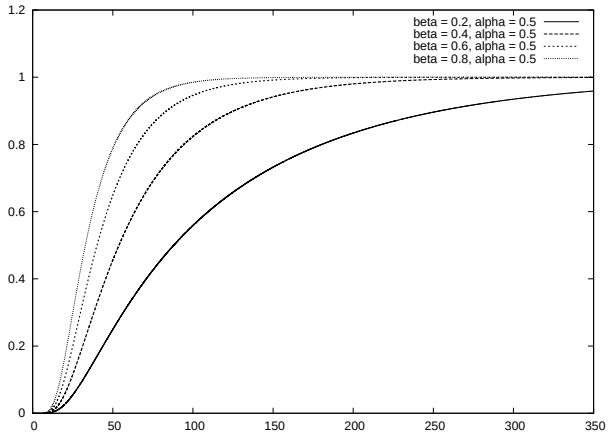


Figure 5. *Probabilité $p(0,t)$ pour le point initial $x_0 = 10$, l'intervalle de diffusion $x \in [0,20]$, $\alpha = 0.5$, β variable* ($\beta = -0.2, -0.4, -0.6, -0.8$)

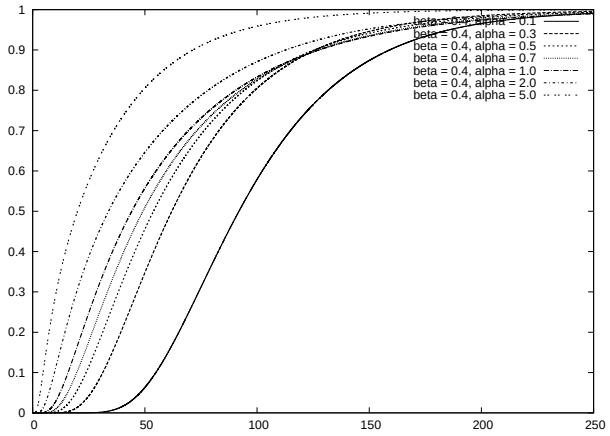


Figure 6. *Probabilité $p(0,t)$ pour le point initial $x_0 = 10$, l'intervalle de diffusion $x \in [0,20]$, $\beta = -0.4$, α variable* ($\alpha = 0.1, 0.3, 0.5, 0.7, 1, 2, 5$)

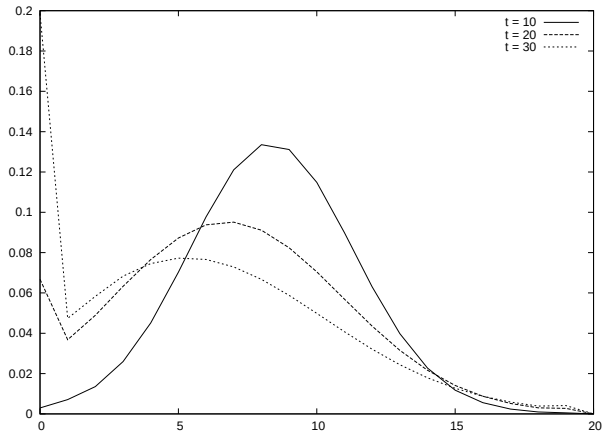


Figure 7. *La densité $f(x,t;x_0)$ de la distance restante à parcourir au paquet aux instants $t = 10, 20$ et 30 pour un réseau caractérisé avec les paramètres $\beta = -0.4$ et $\alpha = 0.54$. Le paquet initialement etait à une distance de $x_0 = 10$ de la destination.*

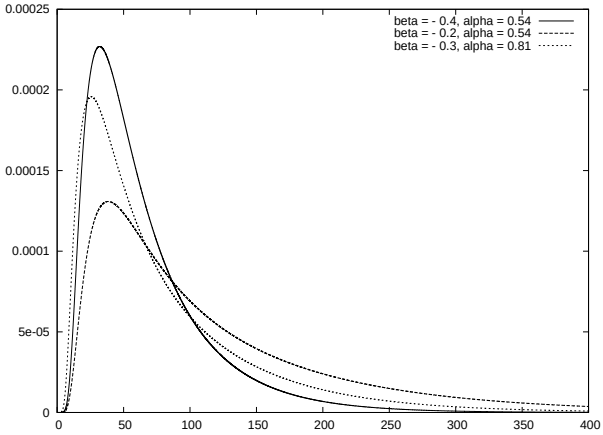


Figure 8. *La densité $\gamma_{x_0,0}(t)$ du temps du premier passage entre x_0 et $x = 0$, qui est l'approximation du temps de transmission entre le point du départ et la destination. Le paquet initialement etait à une distance de $x_0 = 10$ de la destination.*

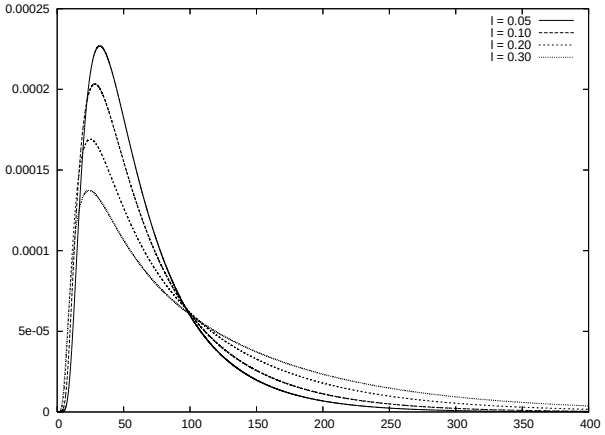


Figure 9. *L'influence de taux de pertes* l $(l = 0.05, 0.1, 0.2, 0.3)$ *sur les densités du temps d'acheminement de paquet pour le réseau de paramètres* $\beta = -0.4$ *et* $\alpha = 0.54$. *Le paquet initialement etait à une distance de* $x_0 = 10$ *de la destination.*

7. Bibliographie

[AKY 02] AKYILDIZ I. F., SU W., SANKARASUBRAMANIAM Y., ÇAYIRCI E., «A survey on sensor networks», *IEEE Commun. Mag.* vol. 40, n° 8, p. 102-114, 2002.

[COX 65] COX R. P., MILLER H. D., *The Theory of Stochastic Processes*, Chapman and Hall, London 1965.

[GEL 75] GELENBE E., «On Approximate Computer Systems Models», *J. ACM*, vol. 22, n° 2, 1975.

[GEL 79] GELENBE E., «Diffusion approximations : waiting times and batch arrival», *Acta Informatica* vol. 12, p. 285-303, (1979).

[GEL 06] GELENBE E., «Travel delay in a large wireless ad hoc network», 2nd Workshop on Spatial Stochastic Models of Wireless Networks (SPASWIN), Boston, 7th April 2006.

[GEL 07] GELENBE E., «A Diffusion Model for Packet Travel time in a Random Multihop Medium», *ACM Trans. on Sensor Networks*, vol. 3, n° 2, Article 10, June 2007.

[HEI 00] HEINZELMAN W. R., CANDRAKASAN A., BALAKRISHNAN H., «Energy-efficient Communication Protocols for Wireless Microsensor Networks», *Proc. Hawaaian International Conf. on Systems Science*, January 2000.

[ROY 99] ROYER E., TOH, C., «A review of current routing protocols for ad-hoc mobile wireless networks"» *IEEE Personal Commun.* vol. 6, n°2, p. 46-55, 1999.

[ZOR 03] ZORZI M., RAO R., «Geographic random forwarding (GERAF) for ad hoc and sensor networks : Multihop performance», *IEEE Trans. Mobile Comput.* vol. 2, n° 4, 337-348, 2003.

QoS et Mobilité dans un système DVB-S/ RCS

Baptiste Jacquemin* — Pascal Berthou*, — Thierry Gayraud*,** — Cédric Baudoin*****

*LAAS-CNRS Université de Toulouse.
7, avenue du Colonel Roche, 31077 Toulouse Cedex 4, France
E-Mail : bjacquem@laas.fr, berthou@laas.fr, gayraud@laas.fr

** Université Paul Sabatier.
118, Route de Narbonne, 31062 Toulouse Cedex 9, France

*** Thales Alenia Space.
26, avenue J.F. Champollion
B.P. 1187, 31037 Toulouse Cedex 1, France
E-Mail : cedric.baudoin@thalesaleniaspace.com

RÉSUMÉ. *Depuis quelques années, les systèmes satellites DVB-S/RCS se révèlent complémentaires aux réseaux terrestres de part leur capacité à offrir un accès large bande pour les régions isolées ou à faible densité sans toutefois nécessiter d'infrastructures lourdes et couteuses. Il devient alors nécessaire que ces systèmes puissent offrir un support approprié aux nouvelles applications multimédias et aux nouveaux services proposés sur les réseaux terrestres tout en tenant compte des contraintes spécifiques qu'ils soulèvent. Dans ce contexte, cet article a pour objet d'examiner plus spécifiquement les problématiques liées à la mobilité d'un utilisateur et à la qualité de service de ses applications dans le cadre d'un système DVB-S/RCS. Une solution basée sur le protocole d'initiation de session (SIP) permettant une gestion couplée de la mobilité d'un utilisateur et de la qualité de service de ses sessions SIP sera ensuite proposée puis évaluée sur une plateforme d'émulation d'un réseau par satellite.*

ABSTRACT. *For a few years, DVB-S/RCS satellite systems have been considered as a complementary solution to terrestrial networks as they are now able to provide broadband access to low populated areas without any wide and costly infrastructure deployment. These networks must be compliant with new requirements such as multimedia applications needs and next generation network services. In such a context, this paper deals with the issues linked to a mobile user who needs Quality of Service (QoS) to run his applications in a DVB-S/RCS system. A solution based on the Session Initiation Protocol (SIP) allowing to manage both mobility and SIP sessions QoS at the same time is then proposed and evaluated thanks to a satellite emulation testbed.*

MOTS-CLÉS : *Qualité de service, DVB-RCS, Mobilité, SIP*

KEYWORDS: *Quality of service, DVB-RCS, Mobility, SIP*

1. Introduction

Les systèmes satellites DVB-S/RCS (Digital Video Broadcasting via Satellite / Return Channel via Satellite) ont, depuis quelques années, réalisé des avancées importantes dans le domaine des transmissions et techniques de codage ainsi que dans la mise en oeuvre d'une voie de retour par satellite offrant ainsi un accès bidirectionnel aux utilisateurs. Cela leur a permis de devenir compétitifs en tant que réseaux d'accès pour les régions isolées ou à faible densité ainsi que pour les réseaux d'entreprise car ils ne nécessitent pas d'infrastructures lourdes et coûteuses. Toutefois, il devient important pour ce type de réseaux de pouvoir offrir à leurs utilisateurs les mêmes services que ceux proposées par le terrestre. En effet, l'apparition de nombreuses applications multimédias (VoIP, e-learning …) et la mobilité de plus en plus grande des utilisateurs ont soulevé de nouvelles contraintes.

De nombreuses études ont alors été entreprises pour faire face à cette demande croissante. Cependant, les solutions proposées pour les réseaux terrestres ne se révèlent pas toujours appropriées aux contraintes spécifiques des systèmes satellites qui souffrent de longs délais de propagation, d'une bande passante limitée et d'erreurs de transmissions plus fréquentes. La mise en œuvre d'une architecture adaptée devient alors cruciale au développement de ces réseaux et cela particulièrement en terme de qualité de service car les contraintes de temps imposées par les applications multimédias ajoutées aux difficultés soulevées par la gestion de la mobilité doivent être prises en compte.

Dans un premier temps, cet article présentera l'architecture globale d'un système satellite DVB-S/RCS ainsi que l'architecture de qualité de service développée au cours de différents projets européens (actuellement SATSIX [1]). Dans un deuxième temps, il examinera la problématique de mobilité dans les systèmes satellites en expliquant les principaux types de mobilité et en justifiant le positionnement de notre étude par rapport à ceux-ci. Ensuite, la partie suivante détaillera notre architecture basée sur le protocole d'initiation de session (SIP) [2] qui permet la gestion de la mobilité pour les applications multimédias tout en assurant une qualité de service appropriée. Enfin, l'implémentation de cette architecture fera l'objet d'évaluations et de validation sur une plate forme d'émulation satellite et sera comparée au protocole Mobile IPv6 [3] dans le cadre d'applications multimédias.

2. Architecture d'un système satellite DVB-S/RCS

2.1. Architecture physique

En 1999, l'ETSI proposa un standard pour la voie de retour par satellite nommé DVB-RCS [4]. Couplée avec la norme DVB-S [5], devenue le standard de diffusion de la télévision numérique par satellite, elle offre un accès bidirectionnel par satellite ouvrant la voie à une nouvelle technologie de réseaux d'accès.

Sans décrire exhaustivement chaque élément du réseau, la figure 1 présente un réseau d'accès satellite à topologie maillée et son intégration au sein d'une infrastructure de type Internet.

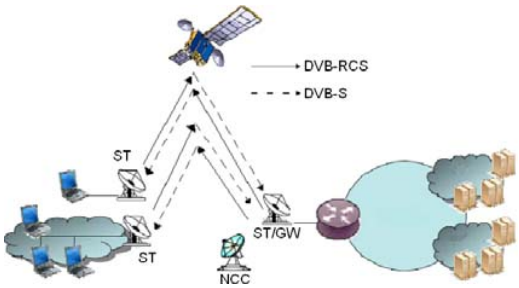


Figure 1. Infrastructure d'accès satellite bidirectionnel

Contrairement à la topologie étoilée dans laquelle tout le trafic est centralisé au niveau de la gateway (GW), l'architecture maillée permet des communications directes entre utilisateurs connectés à des Terminaux Satellites (STs) appartenant à un même réseau satellite. Ces STs ou « satbox » jouent le rôle de routeur d'accès pour le réseau satellite. Un Network Control Center (NCC) est alors en charge du contrôle d'accès aux ressources en partageant le lien de retour entre les différents STs. Dans la suite de l'article, nous considérerons une architecture maillée.

2.2. Gestion de la qualité de service

Un des aspects fondamentaux de la gestion de qualité de service dans ce type de réseau réside dans la gestion de l'accès à la voie montante (ST->satellite), la voie descendante (satellite->ST) étant plus simple à gérer puisque contrainte par ce qui est déjà « monté » et surdimensionnée (jusqu'à 300 fois la voie montante/spot).

L'architecture de QoS développée par le groupe Broadband Satellite Multimédia (BSM) de l'ETSI et normalisée par le SatLab (Association Internationale pour le développement de la technologie DVB-RCS), repose sur l'implantation de services différentiés dans le terminal satellite. La différentiation réside dans la quantité et la périodicité des requêtes de bande passante effectuées auprès du NCC. Un ST est donc un routeur diffserv que l'utilisateur sollicite généralement par le marquage de ses flux applicatifs.

Afin de faciliter l'utilisation de ces services par l'utilisateur et d'améliorer l'adéquation de ses besoins applicatifs à ces services, nous avons proposé plusieurs solutions qui supplantent le simple marquage. Le « QoS agent » permet à l'utilisateur un contrôle visuel des applications et des services associés mais nécessite une certaine expertise. L'extension d'un proxy SIP déployé au niveau de

chaque ST permet un contrôle fin des besoins applicatifs, le tout de manière transparente, en interceptant les requêtes SIP qui contiennent une description précise des flux véhiculés afin d'optimiser la réservation de ressources [6]. Cette dernière solution est une réelle avancée, car la signalisation SIP est principalement utilisée par les applications multimédias qui sont aussi les plus exigeantes en termes de qualité de service. Le « QoS agent » et le proxy SIP communiquent leurs informations à une entité située au niveau des STs, le « QoS server » qui configurera automatiquement la pile diffserv pour fournir la qualité de service adéquate aux flux signalés.

3. La mobilité dans un système satellite

La mobilité dans un système satellite revêt de nombreuses formes qui relèvent de solutions très différentes. Ce paragraphe aborde ces différents types de mobilité au travers de scénarios et permet de situer la contribution de cet article.

3.1. Mobilité des Terminaux Satellites (ST)

La mobilité de terminal satellite (ou « satbox »), qui pourrait sembler de prime abord complexe, trouve sa raison d'être dans de nombreux scénarios de transport. Citons l'exemple du déploiement de l'accès Internet dans les trains à grande vitesse Français [7] qui a débuté en 2007 et qui utilise conjointement des réseaux par Satellite et des réseaux UMTS (dans les gares) couplés à un accès WiFi. Un terminal satellite DVB-S/RCS assure la connexion du train à l'Internet lors de son déplacement. Grâce à la large couverture d'un spot satellite, la mobilité du terminal est transparente pour le réseau lors des déplacements à l'échelle nationale (seule l'antenne embarquée dans le train doit être repositionnée régulièrement en direction du satellite). Lors de plus long trajet, où plusieurs spots peuvent être traversés, des solutions de mobilité existent toutefois au travers de l'extension du protocole DVB-RCS [8]. La mobilité du terminal satellite, et donc des utilisateurs qui y sont raccordés peut être vue comme une mobilité gérée au niveau 2 par le réseau satellite et sort du cadre de cet article.

3.2. Micro mobilité des utilisateurs

Le terme micro-mobilité est utilisé dans le cas d'une mobilité intra-technologie ou dans le cas d'une mobilité intra-domaine (éventuellement inter-technologies). Dans le cas d'un système satellite, on considérera que la micro-mobilité désigne un déplacement derrière un même Terminal Satellite.

Le déploiement d'un réseau WiFi dans une entreprise interconnectée par satellite est un cadre type à ce genre de mobilité. L'utilisateur est mobile dans l'entreprise, mais toute la mobilité est gérée au niveau 2 par les bornes WiFi elles-mêmes. Dans le cas d'une mobilité inter technologies, par exemple lors d'un déplacement entre deux sous réseaux de technologies différentes de l'entreprise, il peut y avoir

changement d'adresse réseau ce qui nécessitera le support de protocoles de mobilité de niveau IP ou supérieur. Dans ces deux cas, l'impact sur la gestion de la qualité de service est limité à l'extérieur du réseau satellite.

3.3. Macro mobilité des utilisateurs

Le terme macro-mobilité désigne une mobilité inter-domaines et éventuellement inter-technologies. Dans le cadre d'un système satellite, cela désigne un déplacement inter-STs.

Mixant mobilité de terminal satellite et mobilité d'utilisateurs, les scénarios d'urgence (catastrophe, incendie) font intervenir de nombreux acteurs sur un terrain où l'infrastructure réseau n'existe pas. Chaque acteur (pompier, gendarmerie) peut posséder son propre accès satellite (dans un camion par exemple) pour déployer un accès réseau sans fil rapidement. Durant l'opération, les acteurs peuvent utiliser les différents moyens de communication qu'ils ont à leur disposition et qui leur sont accessibles, c'est-à-dire potentiellement plusieurs terminaux satellites déployés dans un environnement proche. Dans ce cadre, la qualité de service est un élément crucial et la mobilité doit être gérée par des protocoles de niveau IP ou supérieur.

Le cadre de cet article est de proposer une solution de gestion de la qualité de service couplée à la mobilité, pour ce type de mobilité. Le terme mobilité correspondra donc à la macro-mobilité dans la suite de l'article. Un des objectifs principaux sera, en plus de la gestion de la mobilité à proprement parler, la transparence du basculement des réservations de QoS entre les terminaux satellites concernés.

4. Mobilité et QoS

4.1. Défaut de Mobile IP pour les applications multimédias

L'Internet n'a pas été conçu avec un souci premier de mobilité, ni dans un souci de qualité de service. Ce n'est que plus tard, quand sont apparus des terminaux mobiles, que le protocole IP fut enrichi du support de la mobilité avec Mobile IP. L'objectif de ce protocole distinct du protocole IP, est de masquer la mobilité d'un équipement à ses correspondants sur le reste de l'Internet et d'éviter les ruptures des communications pendant les déplacements. Cependant, Mobile IP ne permet pas de gérer la qualité de service des communications et bien que cela n'ait que peu de conséquences dans le cadre d'applications classiques telles que le transfert de fichiers ou le « web browsing », ce n'est pas le cas pour les applications multimédias temps réel qui ont de fortes contraintes de temps et qui, dans un contexte de mobilité, nécessitent une adaptation au support courant (changement de codecs, éventuellement suppression/ajout de la vidéo selon bande passante disponible).

Il devient donc nécessaire de mettre en place de nouveaux mécanismes pour permettre la gestion de la QoS pour un utilisateur mobile. Deux solutions se révèlent alors possibles : l'ajout de mécanismes au protocole Mobile IP ou la mise en œuvre d'une autre solution basée sur un ou plusieurs autres protocoles.

Etant donné l'architecture de QoS de notre système DVB-S/RCS basée sur les proxies SIP, nous avons choisi d'étudier et d'implémenter une architecture de mobilité à qualité de service basée sur SIP. Certaines études [9] ont déjà montré comment utiliser ce protocole pour traiter la mobilité d'un utilisateur en cours de communications. Le couplage avec la QoS implique alors la spécification de nouveaux échanges entre les différentes entités du réseau (proxy SIP, ST) responsables de la gestion de QoS. Par exemple, lors du déplacement d'un utilisateur, toutes les ressources qu'il mobilise dans son ST devront être transférée et éventuellement modifiés au nouvel ST de rattachement.

4.2. Architecture SIP pour la gestion couplée de la mobilité et de la QoS

4.2.1. Introduction à SIP

Le protocole SIP est en passe de devenir un pilier de ce que pourrait être une future couche session de l'Internet, réunissant autour de lui un consensus autant dans l'Internet que dans les systèmes 3G. Dissociant une adresse de niveau session, qui est un identifiant de ressource universel (URI) du type (sip:kiaso@domaine.org), de l'adresse de niveau réseau (adresse IP), le protocole SIP permet, outre la gestion de la session, la localisation d'un utilisateur mobile.

De part son architecture basée sur différents serveurs qui permettent l'enregistrement, la localisation d'un utilisateur ainsi que le relais des messages par les proxy, SIP permet aisément la gestion de la mobilité nomade dans laquelle un utilisateur change de point d'accès avant d'initier ou de recevoir une invitation de session. L'objectif de notre étude a alors été de concevoir et d'étudier une solution basée sur SIP permettant la gestion couplée de la mobilité d'un utilisateur en cours de communications et de la qualité de service de ses communications multimédias. Dans le paragraphe suivant, nous allons donc décrire notre proposition en détail.

4.2.2. Présentation de notre architecture

Notre architecture sera basée sur le JAIN SIP Presence Proxy [10] ainsi que sur un client SIP développé au LAAS. Il est tout d'abord important de préciser que, pour notre architecture, les proxies SIP doivent fonctionner en mode « statefull », c'est-à-dire qu'ils conservent en mémoire l'état des transactions (méthodes et réponses finales). De plus, des modifications par rapport à la norme ont été réalisées pour **forcer les messages SIP à passer par les proxies** situés au niveau de chaque terminal satellite concerné par la communication, permettre l'envoi de messages de **réservation/libération de QoS** au «QoS server» et permettre **la gestion optimisée du réenregistrement** dans un contexte de mobilité nomade et continue en tenant compte des contraintes spécifiques du satellite.

Nous allons d'abord insister sur ce point dans la partie 4.2.2.1 car c'est à ce niveau que les principales améliorations du proxy et du client SIP ont été apportées. Enfin, en partie 4.2.2.2, nous présenterons la méthode implémentée au niveau du client pour reprendre ses communication avec un temps d'interruption minimum.

4.2.2.1. Problématique du réenregistrement

On considère une communication entre un nœud mobile (MN) et un correspondant (CN). Une session SIP a permis de réserver des ressources au niveau des STs concernés. Une fois que le MN a changé de réseau et obtenu une nouvelle adresse, il doit tout d'abord effectuer la procédure de réenregistrement/libération de ressources qui se décompose en 2 ou 3 phases selon les cas:

– Prévenir son home proxy SIP qu'il a changé d'adresse pour qu'il puisse continuer à être joignable de part son URI SIP. D'autre part, si la QoS de la communication avait été réservée au niveau de ce proxy, elle doit être libérée.

– S'enregistrer auprès du proxy SIP qui s'occupe du nouveau domaine pour que les requêtes soient correctement transmises.

– Se désenregistrer auprès du dernier SIP proxy si celui-ci n'était pas le « home » SIP Proxy. Il est alors aussi nécessaire de libérer la QoS allouée.

Pour ces enregistrements/désenregistrements, il est soit possible d'envoyer un message pour chaque proxy SIP, soit d'envoyer un seul message à destination du proxy SIP en charge du domaine courant et ensuite, celui-ci se chargera de prévenir les autres proxies. Dans le cadre du nomadisme, la première solution n'est pas pénalisante puisque l'utilisateur n'est pas en cours de communication et donc son temps d'enregistrement/désenregistrement n'est pas important. Par contre, dans le cadre d'une mobilité continue et puisque l'on force tous les messages SIP à passer par tous les proxies, l'utilisateur doit se réenregistrer dans son nouveau domaine avant de pouvoir relancer ses communications et donc, si l'utilisateur envoie un message REGISTER qui doit traverser le satellite, le temps d'interruption des communications sera allongé d'un aller retour à travers le satellite (minimum 500 ms).

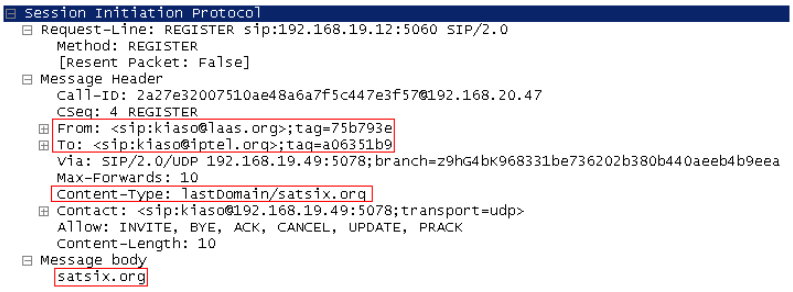


Figure 2. *Format spécial du message reREGISTER*

Nous avons donc choisi d'implémenter la deuxième solution qui permet d'une part un gain de temps et d'autre part d'éviter l'envoi multiple de messages REGISTER (résumé sur la figure 4). La figure 2 montre le format d'un message REGISTER créé par notre client SIP dans le cas où un utilisateur avec comme URI « kiaso@laas.org » (laas.org correspond au domaine mère de l'utilisateur) passe d'un réseau visité 1 (dans notre exemple satsix.org) à un autre réseau visité 2 (iptel.org). Il doit donc s'enregistrer auprès de son nouveau proxy SIP, prévenir celui de son réseau mère et se désenregistrer auprès de son dernier proxy.

Pour différencier un enregistrement classique d'un enregistrement avec nécessité de prévenir un réseau mère, nous avons utilisé la possibilité d'avoir un champ « From » différent du champ « To » dans le message REGISTER comme autorisé par la norme [2]: «The From header field contains the address-of-record of the person responsible for the registration. The value is the same as the To header field **unless the request is a third-party registration** ». On considère ici que l'utilisateur, bien qu'il soit la même personne physique, représente une « third party » vu qu'il a changé de domaine et d'adresse. Ainsi, le proxy SIP voit arriver un message REGISTER avec un champ « From » comportant l'adresse « user@DomaineMere.org » (dans la figue ci-dessus kiaso@laas.org où laas.org représente le domaine mère) et un champ « To » avec l'adresse « user@DomaineVisité2.org » (kiaso@iptel.org ci-dessus où iptel.org représente le domaine courant ou domaine visité2). En voyant ces deux champs différents, ce proxy envoie automatiquement une requête DNS de type SRV permettant de connaître le proxy en charge du domaine correspondant au champ « From » et prévient ce dernier que c'est maintenant lui qui se charge des requêtes de l'utilisateur mobile. Le proxy en charge du réseau mère transmettra alors toutes les requêtes à destination de « user@DomaineMere.org » vers « user@DomaineVisité2.org ».

Enfin, pour prévenir l'ancien proxy SIP qui doit maintenant désenregistrer l'utilisateur mobile et libérer les ressources qui lui étaient allouées, on a utilisé le champ «Message Body » inutilisé pour les messages REGISTER pour y inscrire le nom de domaine correspondant au dernier réseau visité, dans notre cas «DomaineVisité1.org ». Le contenu de ce message Body est décrit dans le champ « Content-Type » encadré en rouge sur la figure 2 et correspondant dans notre cas au lastDomain/satsix.org où satsix.org représente le dernier domaine visité.

4.2.2.2. Détection de changement de réseau

En plus des modifications réalisées au niveau de l'enregistrement, il a fallu permettre au client de détecter automatiquement un changement de réseau en lui ajoutant un module qui permet, lors d'un changement de réseau, de réaliser les différentes étapes suivantes:

– Détecter que l'adresse IP de l'utilisateur a changé.

– Connaître le nouveau domaine courant.

– Envoyer une requête DNS de type SRV (_sip._udp.iptel.org dans le cas de l'exemple précédent) pour connaître le proxy SIP en charge de ce domaine.

– Réaliser la procédure de réenregistrement décrite dans la partie précédente.

– Ré initier la session SIP à partir de la nouvelle position.

La figure 3 montre le format du message reINVITE qui est en fait le même que le message INVITE initial sauf que l'adresse comprise dans les champs entourés en rouge est maintenant la nouvelle adresse IP de l'utilisateur. Par contre, il est important de ne pas modifier l'adresse comprise dans le callID qui correspond à l'adresse initiale de l'utilisateur et qui identifie la session.

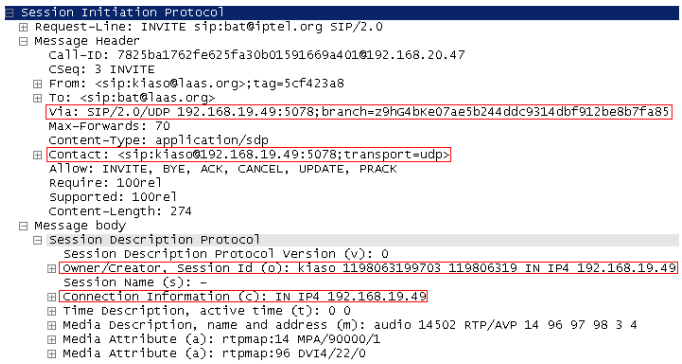


Figure 3. *Format spécial du re-INVITE*

Les descripteurs SDP peuvent également être modifiés (type de codec, ordre des codecs …) pour adapter la qualité de service au nouveau support.

La figure 4 résume les différentes étapes permettant de retrouver une communication SIP et gérer la qualité de service au niveau du système satellite. Une fois que le nœud mobile a changé de réseau (2) alors qu'il était en cours de communication (1), il s'enregistre (3) auprès du proxy SIP 2 en charge de ce domaine qui va maintenant prévenir les proxies concernés :

–A : Le proxy 2 envoie un message au proxy 1 pour lui dire de désenregistrer MN

–B : Le proxy 1 libère alors les ressources associées au MN en envoyant un message FREE au « QoS server ».

–C : Le proxy SIP 2 envoie un message au home proxy SIP pour mettre à jour la localisation du MN.

–D : Le home proxy SIP libère les ressources précédemment associées au MN.

Parallèlement, le client SIP peut alors ré-initier la session (4) et la QoS sera réallouée au niveau des nouveaux proxies en fonction des nouveaux paramètres.

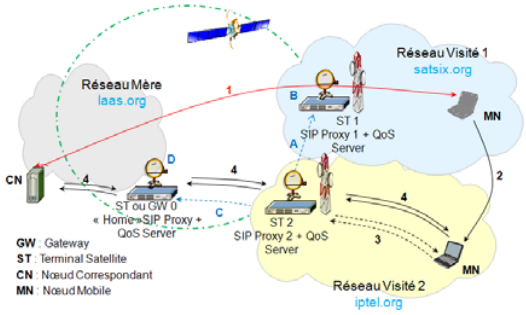


Figure 4. *Gestion couplée de la mobilité et de la QoS par SIP*

5. Evaluation et validation de l'architecture proposée

Dans cette partie, après avoir brièvement présenté la plate forme d'émulation d'un système satellite qui nous permettra de tester notre architecture, nous évaluerons les temps d'interruption pour notre solution basée sur SIP et utilisant IPv6 en la comparant à une solution utilisant le protocole Mobile IPv6.

5.1. Plateforme d'émulation de réseau satellite DVB-S/RCS

Pour évaluer l'architecture que nous avons mise en œuvre et la comparer avec Mobile IPv6, elle a été implémentée et testée sur une plateforme d'émulation satellite de type DVB-RCS. Basée sur un réseau de 10 machines indépendantes, elle reproduit les mécanismes d'encapsulation, les techniques d'accès au canal satellite (allocation de bande passante à la demande), l'émulation des canaux montant et descendant et les caractéristiques de transmission spécifiques au satellite (délai, gigue, profil d'erreur). Seul le cœur du système (satellite, gateway, ncc, STs) est spécifique au système d'émulation, les terminaux utilisateurs quand à eux sont des systèmes classiques reliés aux terminaux satellites (ST) par IP.

5.2. Temps d'interruption : comparaison entre Mobile IPv6 et Mobilité SIP

On considère initialement une communication entre un nœud mobile qui se trouve dans son réseau mère situé derrière le ST0 et un correspondant qui se trouve dans un réseau que nous appellerons « réseau du correspondant » situé derrière le ST1. Le « home agent » nécessaire à Mobile IP se trouve dans le réseau mère. On utilise ici les mêmes noms (ST0, ST1 …) que dans la figure 4 mais on considère le cas plus général où le CN ne se trouve pas dans le réseau mère. Les différents déplacements inter-STs du nœud mobile considérés dans le tableau précédent sont:

–Le déplacement 1 : depuis le réseau mère (ST0) vers un réseau qui n'est pas celui de son correspondant (ST2).

–Le 2 : depuis ce dernier réseau vers le réseau du correspondant (ST1).

–Le 3 : depuis ce dernier réseau vers le réseau mère (ST0).

Dans l'article [11], on voit que la mobilité SIP permet théoriquement un gain de temps par rapport à Mobile IPv6. Nous allons maintenant nous focaliser sur les temps expérimentaux. Pour Mobile IPv6, on utilise la procédure de « return routability test » et on considère qu'il y a interruption jusqu'à ce que l'optimisation de route ait été réalisée car pendant la phase d'optimisation, une communication VoIP est beaucoup trop saccadée (sauf dans le cas du déplacement 3 où le mobile revient dans son réseau mère). De plus, on considère les temps d'interruption sans les mécanismes de détection d'adresse dupliquée qui ajoute entre 1.25 et 1.75 s.

	Mobilité SIP avec DNS (et sans)	Mobile IPv6
déplacement 1	2.11 s (1.10 s)	3.97 s
déplacement 2	2.12 s (1.10 s)	1.02 s
déplacement 3	2.56 s (1.53 s)	0.65 s

Tableau 1. *Temps d'interruption correspondants aux différents déplacements*

On peut d'abord constater que lorsque l'on utilise une découverte dynamique du proxy SIP courant (requête DNS de type SRV), cela rajoute environ un temps de traitement de 1 s au niveau du client SIP ce qui est pénalisant. Lorsque le client a déjà une liste des proxies associés à leur domaine en mémoire, le temps d'interruption est donc fortement réduit. En comparant les résultats sans DNS avec Mobile IPv6, Pour le déplacement 2, les temps sont à peu près similaires mais par contre, dans le cas d'un retour dans le réseau mère (déplacement 3), Mobile IPv6 se révèle plus efficace de part son architecture construite autour du « home agent ». Cependant, on constate que, pour le déplacement 1 qui est le plus courant, SIP est beaucoup plus efficace que Mobile IPv6 dont la phase d'optimisation est très longue. Donc, même si dans quelques cas particuliers, le temps d'interruption est plus faible pour Mobile IPv6, la mobilité SIP se révèle plus efficace dans le cas général avec ou sans découverte dynamique des proxies SIP.

5.3. Validation de la QoS dans le cadre de la mobilité

Lors des expérimentations, on a pu valider la libération des ressources allouées pour la communication avant déplacement ainsi que la réservation de nouvelles ressources au niveau des STs concernés par la communication après déplacement. Le message de réservation contient des données sur les codecs utilisés (débit, perte max, délai max, Call ID, type de média, ports, adresses …) qui permettront au STs de demander au NCC de lui allouer une quantité appropriée de bande passante.

6. Conclusion

En conclusion, notre architecture SIP basée sur une communication inter-proxies et sur un client amélioré pour la mobilité présente l'avantage d'offrir une gestion

couplée par un même protocole de la QoS et de la mobilité d'un terminal dans le cadre d'applications multimédias tout en permettant d'obtenir des temps d'interruption moins importants que ceux obtenus avec Mobile IPv6 dans le cas général. Cette diminution du temps d'interruption ainsi que la gestion automatique de la réservation/libération de QoS au niveau des ST sont particulièrement importantes dans un système satellite qui veut offrir à ses utilisateurs des applications multimédias. De plus, l'architecture SIP est valable aussi bien en IPv4 qu'en IPv6 ce qui lui permettra de s'adapter à n'importe quels types de réseaux dans le futur.

Ces travaux laissent entrevoir un certain nombre de perspectives. Tout d'abord, des évaluations complémentaires doivent être réalisées au niveau de la mobilité et de la QoS ainsi que sur la charge des proxies SIP. Ensuite, il serait intéressant de coupler notre architecture SIP qui ne permet que de gérer la mobilité pour des applications multimédias à Mobile IPv6 (ainsi que ses extensions) qui reste la meilleure alternative pour les applications classiques basées sur TCP. On pourrait aussi étudier un couplage de Mobile IPv6 avec le "QoS agent" pour qu'un utilisateur puisse gérer la QoS de ces autres applications (par exemple, s'il veut contrôler le débit de ses téléchargements). Enfin, les aspects concernant la sécurité et l'authentification devront être traités pour assurer qu'un utilisateur malicieux ne s'approprie pas les ressources d'un autre.

7. Références

[1] IST SATSIX Project (Contract IST-2004-26950), http://www.ist-satsix.org
[2] J. Rosenberg, H. Schulzrinne & al. "SIP: Session Initiation Protocol", RFC 3261, June 2002.
[3] D. Johnson, C. Perkins & al."Mobility Support in IPv6", RFC 3775, June 2004.
[4]ETSI: EN 300 421 V1.1.2 (97-08) Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for the 11/12 GHz satellite services.
[5] ETSI: ETSI EN 301 790 V1.4.1 (2005-04) Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems.
[6] O. Alphand, P.Berthou, T.Gayraud, S. Combes, « Signalisation de QdS dans un réseau satellite de nouvelle génération », CFIP 2005, Bordeaux, Mars 2005.
[7] Commniqué de presse, 21Net, http://www.21net.com/EN/pdf/fr-system-2007.pdf,
[8] G. Acar, K. Kasparis, P. Thomson, « The Enhancement of DVB-S2 & DVB-RCS by Adding Additional Mobile User Capability »Digital Video Broadcasting Over Satellite: Present and Future, 2006, 28-28 Nov. 2006
[9] H. Schulzrinne and E. Wedland. "Application-layer mobility using SIP", ACM SIGMOBILE, Vol. 4, Number 3, July 2000, pp.47-57.
[10] https://jain-sip-presence-proxy.dev.java.net/
[11] B. Jacquemin, T. Gayraud et P. Berthou. « Architecture de mobilité dans un système satellite DVB-S/RCS », Journées Doctorales en Informatiques et Réseaux (JDIR'2007), Janv. 2007.

Evaluation des performances réseau dans le contexte de la virtualisation XEN

Pascale Primet* — **Olivier Mornard*** — **Jean-Patrick Gelas****

** INRIA Rhône-Alpes, ** Université de Lyon*
Laboratoire LIP
École Normale Supérieure de Lyon
46, allée d'Italie
69364 Lyon Cedex 07 - France

{pascale.primet,olivier.mornard,jpgelas}@ens-lyon.fr

RÉSUMÉ. *Les techniques de virtualisation des systèmes d'exploitation améliorent la sécurité, l'isolation, la fiabilité et la flexibilité des environnements. Ces techniques deviennent incontournables dans les systèmes distribués large échelle et commencent à être exploitées dans les réseaux. Cependant la virtualisation introduit un surcoût devant être intégré aux modèles de performance des dispositifs virtualisés afin d'en prédire le comportement. Dans cet article nous analysons ce surcoût et évaluons expérimentalement l'impact de la virtualisation* XEN *sur les performances réseau dans le cas des réseaux haut débit. Nous montrons que le débit obtenu par TCP est très peu affecté par la virtualisation au prix d'une augmentation de la consommation CPU. Nous étudions les conflits entre le traitement protocolaire et les processus de calcul pour l'accès aux capacités des processeurs. Nous montrons aussi l'impact de l'ajustement de la taille de la fenêtre de congestion de TCP dans le cas de plusieurs machines virtuelles.*

ABSTRACT. *Virtualization technics of the Operating System improve security, isolation, reliability and flexibility of the environments. These technics become a must in large scale distributed system and begin to be used in data transport networks. However, virtualization introduced an overhead which must be integrated to virtualized system performance models in order to forecast their behavior. In this article, we analyse this overhead and give an experimental evaluation of the virtualization impact with* XEN *over the network performance in high speed network context. We show that the throughput reached with TCP is barely affected by the virtulization at the expense of a growing CPU consumption. Moreover, we study the conflicts between protocol process and standard process to access CPU resources. Finally, we show the impact of the TCP's congestion window size in case of several virtual machines running.*

MOTS-CLÉS : *Performances réseaux, virtualisation, Xen*

KEYWORDS: *Network performances, virtualization, Xen*

1. Introduction

La virtualisation a été proposée par IBM dès le milieu des années 60. L'idée générale est de permettre de lancer de multiples instances de système d'exploitation sur le même matériel. Ainsi chaque instance se comporte fonctionnellement exactement comme si elle était en cours d'exécution seule sur le même matériel. Aujourd'hui, cette approche remporte un immense succès car elle permet d'améliorer la sécurité, l'isolation, la fiabilité et la flexibilité des environnements. Ainsi, ces techniques deviennent incontournables dans les systèmes distribués large échelle dans lesquels ces défis doivent être relevés. Dans le domaine des réseaux, le phénomène est plus récent et plusieurs projets explorent l'intérêt d'utiliser les techniques de virtualisation dans les routeurs eux-mêmes. Les fabriquants de processeurs, quant à eux, fournissent aujourd'hui en standard un jeu d'instructions processeur (virtualisation matérielle) permettant d'isoler efficacement les machines virtuelles, et ce même dans les machines grand public.

Dans le cas des systèmes distribués large échelle, la virtualisation permet la protection contre les programmes malveillants, l'évitement des conflits, la sécurité de l'exécution dans un environnement certifié, la mesure de la consommation de ressources. L'exécution de longs travaux peut être fiabilisée grâce à la possibilité de redémarrer à partir d'un certain état sauvegardé. La virtualisation est très utile quand le redimensionnement de l'environnement en termes de capacités (CPU, mémoire, bande passante d'E/S, etc.) est nécessaire. Enfin la virtualisation permet d'optimiser l'utilisation d'un ensemble de ressources par équilibrage de charge (*load-balancing*) des machines virtuelles sur des machines physiques.

L'utilisation de la virtualisation système dans le cas des routeurs permet de simplifier leur gestion, d'augmenter la sécurité, de protéger certains trafics, d'adapter la gestion des flux en fonction des ressources, de la consommation d'énergie, d'adapter le système et les processus de routage aux trafics et non l'inverse. On peut aussi envisager le partage d'un même équipement par plusieurs organisations. La virtualisation s'avére être aussi une approche très intéressante pour l'expérimentation.

Si la virtualisation offre de nombreux avantages et de nombreuses opportunités, elle apporte néanmoins une contrepartie : son coût non nul et indéterministe en terme de consommation de ressources. Alors que le surcoût et le passage à l'échelle des techniques de virtualisation a été bien étudié au niveau système d'exploitation, peu de travaux ont été menés dans le domaine de l'analyse des performances réseau de bout en bout. Nous nous attachons donc à évaluer et à comprendre le surcoût de consommation de bande passante réseau et de traitement protocolaire, en particulier dans les réseaux haut débit. En effet, ce surcoût doit être intégré aux modèles de performance des dispositifs virtualisés (cluster virtuels ou routeurs virtuels) afin d'en prédire au mieux le comportement. Dans cet article, nous nous focalisons sur l'évaluation de l'environnement *XEN* qui est un logiciel libre très utilisé. La section 2 propose un bref état de l'art des différentes méthodes de virtualisation. Nous développons dans la section 3 une analyse des mécanismes réseau dans le cas de la solution *XEN*. Enfin la

section 4 propose une évaluation des performances réseau obtenues dans le contexte du protocole TCP et de l'environnement *XEN*. La dernière section 6 développe la conclusion et quelques perspectives.

2. Les mécanismes de virtualisation

Avant d'aborder l'étude des interactions entre le réseau et les mécanismes de virtualisation d'OS, nous exposons les différentes approches de virtualisation existantes :

Virtualisation au niveau application : Cette approche permet de virtualiser de multiples serveurs isolés et sécurisés sur un seul serveur physique. Il n'y a pas de système d'exploitation dans les machines virtuelles. Tous les serveurs virtuels partagent le même système d'exploitation : celui de la machine physique. *VServer*[GEL 03] est basé sur cette approche.

Émulateurs : Dans l'approche émulateur, une machine virtuelle est simulée ainsi que tout son matériel spécifique. Il est donc possible d'utiliser un système d'exploitation classique non modifié sur cette architecture. On peut également émuler un processeur différent de celui de la machine hôte. Cette technique est couramment utilisée pour développer des logiciels pour les nouveaux processeurs avant qu'ils ne soient effectivement disponibles.

Système d'exploitation dans l'espace utilisateur : Une autre possibilité est d'exécuter le système d'exploitation dans l'espace utilisateur comme une application classique. Cette solution n'est pas très performante et est principalement utilisée pour les développements des noyaux des systèmes. Le plus célèbre projet basé sur cette approche est *UML* ou *User Mode Linux* [HOX 02].

Paravirtualisation : Par rapport à l'approche précédente, la technique de paravirtualisation ne nécessite pas de simuler le matériel, mais plutôt d'offrir une API offrant les mêmes services que le matériel. Ceci implique une modification du système invité. La série d'abstractions (processeur virtuel, mémoire virtuelle, réseaux virtuels, etc…) permet à différents OS d'être portés. Ces abstractions ne sont pas nécessairement similaires à de réelles ressources matérielles de la machine hôte. Ces services sont fournis par des appels système spéciaux effectués auprès d'un hyperviseur. Le projet le plus connu basé sur cette approche est *XEN* [BAR 03, PRA 05]. *XEN* est un moniteur de machines virtuelles pour l'architecture x86, qui permet des exécutions simultanées de plusieurs OS en fournissant une isolation des ressources et un confinement des différents environnements d'exécution. Dans *XEN*, un domaine 0 (Dom0), créé sur l'ordinateur hôte au démarrage, accède à l'interface de contrôle et exécute l'application de gestion des domaines virtuels. Le domaine 0 permet de contrôler le lancement et l'arrêt des autres domaines dans lesquels les OS invités s'exécutent. *XEN* propose deux ordonnanceurs : «Borrowed Time virtuelle» qui fournit un partage équitable des ressources CPU aux différents domaines, et «Atropos» qui est un ordonnanceur temps réel à contraintes relachées, il autorise un ordonnancement

sur la base du «best effort». Pour la virtualisation de la mémoire, *XEN*réserve pour chaque VM (Virtual Machine) une certaine quantité de mémoire physique. Cette zone mémoire est fixée à l'initialisation de chaque VM, mais elle peut être adaptée si un utilisateur a besoin de plus ou moins de mémoire (sans quitter ou redémarrer la VM). L'utilisation de *XEN* n'est pas transparente, puisque les systèmes d'exploitation invités doivent être portés avant d'être installés pour pouvoir fonctionner. *XEN* fournit un haut degré de confinement, avec la virtualisation du CPU et des ressources.

Virtualisation assistée matériellement : Par rapport à l'approche précédente, la virtualisation assistée par le matériel permet d'utiliser un OS invité non modifié[QUM]. La machine virtuelle dispose de son propre matériel virtuel et les OS peuvent être utilisés isolés les uns des autres. Dans de nombreux cas, la machine virtuelle exécute un système d'exploitation différent de celui de l'ordinateur hôte. Cette approche a été rendue récemment possible (2005) grâce à une amélioration du jeu d'instructions des processeurs fournie par Intel (VT Technology) et AMD (Pacifica), mais aussi par IBM (IBM advanced POWER virtualisation) et Sun (Sun UltraSPARC T1 hypervisor). L'objectif de cette démarche est de faire tourner les OS invités non modifiés sans perte de performances.

3. Virtualisation de l'accés au réseau

3.1. *Protocoles et accès réseau classiques*

Sous Unix, et plus spécifiquement dans sa version Linux, les applications communiquent à travers l'interface *socket* qui permet l'accès aux services de transport (TCP, UDP, DCCP, SCTP,...) implantés par les protocoles de transport du noyau. Ces protocoles interagissent avec les composants du protocole IP qui assurent l'acheminement des paquets à travers les différents équipements intermédiaires jusqu'au destinataire. La couche IP dialogue avec la carte d'interface réseau (typiquement Ethernet) via son pilote (driver ou module) situé dans le noyau et représenté par une "interface". A l'émission, le système copie les données dans une zone spéciale, le socket buffer (`skbuf`), puis rend la main à l'application. Pendant ce temps, le système d'exploitation se charge d'envoyer les données en arrière plan de cette zone vers la carte d'interface réseau par un accès direct à la mémoire (DMA, *Direct Memory Access*). Ainsi l'application n'est bloquée que pendant la première copie, au lieu d'attendre la réception du dernier message d'acquittement. En réception, la stratégie est symétrique, mais le récepteur est vraiment bloqué jusqu'à l'arrivée des données en mémoire utilisateur. Ce modèle a pour inconvénient d'être gourmand en temps processeur et en occupation du bus mémoire (3 accès côté émetteur et 3 côté récepteur). Ainsi, les communications à haut débit consomment un très grande quantité de temps CPU (les traitements protocolaire d'une connexion à 10Gb/s saturent un ordinateur moderne). Le système d'exploitation fournit plusieurs fonctions pour le traitement de la pile protocolaire : l'ordonnancement des tâches liées au protocole, la gestion de la mémoire, la gestion

des interruptions et les opérations d'accès au matériel. L'ordonnancement du protocole a une grande influence sur l'efficacité et les performances temps-réel du protocole.

Au niveau des performances que l'on peut attendre en terme de débit bout en bout entre les deux extrémités d'une connexion Ethernet, deux valeurs peuvent être considérées : le débit émis et le débit utile reçu. Considérons une liaison physique de type Ethernet. Soit R_{EthB} le débit binaire Ethernet. $R_{\text{EthB}} = 10^9 bps$ pour du Gigabit Ethernet. Nous définissons R_{Eth} le débit trame Ethernet en faisant l'hypothèse de trames de 1514 octets (sans le checksum (FCS), le préambule et l'*inter frame gap* (IFG)). $R_{\text{Eth}} = (1514/(1514 + 4 + 12 + 8)) * R_{\text{EthB}} = 984.40$Mbps Nous définissons R_{IP} le débit IP paquets (Ethernet payload). Avec l'hypothèse d'une MTU [1] de 1500 octets : $R_{\text{IP}} = (1500/1514) * R_{\text{Eth}} = 975.30$Mbps.

Le débit d'un protocole de transport TCP idéal sur un lien 1Gbps Ethernet (sans le *slow start* ni le mécanisme de retransmissions de TCP) est défini par R_{TP}. Sous les hypothèses précédentes, $R_{\text{TP}} = ((1500 - 52)/1500) * R_{\text{IP}} = 941.49$Mbps Ainsi cette valeur représente le *goodput* (R_{TP}) et (R_{EthB}) le débit émis effectivement sur le lien. Ainsi le débit utile maximum atteignable sur un lien gigabit Ethernet est donc de 941.49Mbps.

En termes de temps de calcul et d'utilisation processeur, nous proposons le modèle suivant qui considère les différentes étapes de traitement dans le noyau et de l'interface (côté émetteur et côté récepteur). Soient : $C_{\text{driversnd}}(S))$ le coût de transmission, fonction de la taille (S) des données à transmettre ou de réception ($C_{\text{driverrcv}}(S)$) sur la carte (copie), C_{IP_*} le coût du traitement IP en émission ou en réception, C_{TCP_*} le coût de traitement TCP en émission ou en réception, $C_{\text{inet}_*}(S)$ le coût de copie dans/vers l'espace utilisateur (fonction de la taille du message à transmettre), et enfin C_{syscall_*} le coût de l'appel système. Ainsi, au niveau de l'émetteur, nous modélisons le coût CPU $C_{\text{TPsnd}}(S)$ comme suit :

$$C_{\text{TPsnd}}(S) = C_{\text{syscallsnd}} + C_{\text{inetsnd}}(S) + C_{\text{TCPsnd}} + C_{\text{IPsnd}} + C_{\text{driversnd}}(S)$$

Au niveau du récepteur, le coût CPU C_{TPrcv} est :

$$C_{\text{TPrcv}}(S) = C_{\text{driverrcv}}(S) + C_{\text{IPrcv}} + C_{\text{TCPrcv}} + C_{\text{inetrcv}}(S) + C_{\text{syscallrcv}}$$

3.2. *Analyse de l'accès réseau dans XEN*

XEN [BAR 03] ne gère pas les périphériques directement, c'est le travail du noyau Linux exécuté en tant que *domaine 0* (Dom0). Les drivers de toutes les cartes réseaux physiques doivent être fournis par le noyau du Dom0. Chaque machine virtuelle possède une ou plusieurs interfaces réseau virtuelle. La communication entre les interfaces est effectuée en utilisant deux buffers circulaires à jetons (un pour l'envoi, un pour la réception). Afin d'assurer un accès fiable aux matériel ainsi que la possibilité de le paramétrer, la virtualisation du matériel dans *XEN* se conforme au modèle *split*

1. Message Transfer Unit

device driver présenté dans [LES 96]. Le Dom0 peut utiliser un dispositif de routage, de pontage (bridge) ou de NAT virtuel pour transmettre les paquets sur les interfaces physiques. Ainsi, au niveau du réseau, le noyau du domaine invité envoie les paquets IP au pilote frontal qui se charge de les transmettre au pilote générique situé dans le domaine 0 via le module netloop (voir figure 1. Ainsi, avec *XEN*, les paquets sont recopiés du socket buffer vers la zone mémoire associée au driver de la carte réseau du domaine 0. Pour gérer les multiples communications issues des différents domaines invités, *XEN* implémente dans le domaine 0 un routeur, un pont ou même NAT. C'est cet équipement virtuel qui ordonnance les paquets et les transmet sur l'interface physique. Dans le cas du pont virtuel, l'arbitrage entre les différentes communications se fait selon une simple politique FIFO, correspondant à l'ordre d'arrivée des paquets dans le driver du domaine 0.

La figure 2 schématise le principe du bridge logiciel proposé pour l'aiguillage des trames issues des différentes machines virtuelles.

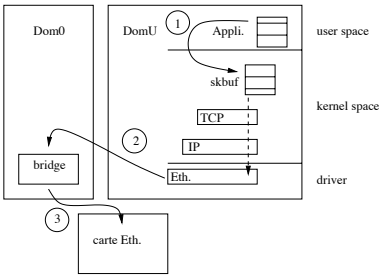


Figure 1. *Le chemin des données et les 3 copies*

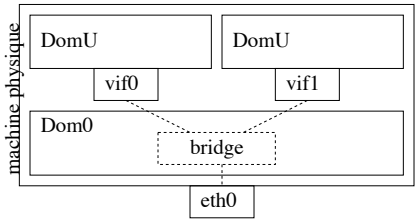


Figure 2. *Bridge logiciel (dans le dom 0)*

Nous proposons donc de compléter le modèle de coût CPU lié aux communications en y intégrant le coût des traitements liés à virtualisation. Au niveau émetteur, nous modélisons C_{TPsndV} :

$$C_{\text{TPsndV}}(S) = C_{\text{TPsnd}}(S) + C_{\text{netloopsnd}}(S) + C_{\text{bridgesnd}} + C_{\text{driversndV}}(S)$$

$$C_{\text{TPsndV}}(S) = C_{\text{syscallsnd}} + C_{\text{inetsnd}}(S) + C_{\text{IPsnd}} + C_{\text{TCPsnd}} + C_{\text{netloopsnd}}(S) + C_{\text{bridgesnd}} + C_{\text{driversndV}}(S)$$

Au niveau du récepteur, le coût CPU C_{TPrcv} est modélisé de manière symétrique. Où $C_{\text{driver*V(S)}}$ correspond au coût de transmission ou de réception sur la carte (copie 1) via le driver générique, C_{bridge_*} au coût de traitement du pont virtuel, $C_{\text{netloop}_{*(S)}}$ au coût de la copie (copie 2) du domaine invité vers le domaine 0 (et réciproquement). On peut s'attendre à ce que la copie entre le domaine invité et le domaine 0 constitue une part importante du coût CPU total de la transmission.

4. Expérimentations et analyse des résultats

L'objectif de ces expérimentations est d'étudier le surcoût de l'implantation de la virtualisation pour l'accès au réseau dans le cas de l'environnement *XEN*. Nous nous concentrons sur les performances de bout en bout obtenues au niveau utilisateur via le protocole TCP. Comme nous l'avons présenté dans la section précédente, la gestion des entrées-sorties réseau dans *XEN* fait intervenir un certain nombre de composants logiciels (en particulier netloop et un équipement virtuel en domaine 0) qui s'intercalent dans le chemin classique des données. Ces composants peuvent être arbitrairement invoqués par l'ordonnanceur des tâches et faire des copies supplémentaires de paquets. Nous avons donc développé un plan d'expériences pour mesurer le coût dû à la virtualisation au niveau du client et au niveau du serveur en termes de débit et de capacité CPU (expérience 1). Le paramétrage de TCP étant relativement délicat dans le cas des réseaux haut débit, nous avons étudié les différentes valeurs de taille de buffer (expérience 2). Enfin, comme la virtualisation permet d'instancier plusieurs machines virtuelles sur une même machine, nous avons cherché à savoir si le partage des ressources était équitable et prévisible (expérience 3).

4.1. *Plate-forme expérimentale et outils*

Les résultats présentés dans cette section sont obtenus sur les équipements mis à disposition dans la plate-forme expérimentale Grid5000[CAP 06]. Il s'agit de serveurs IBM eServer 325 dotés de processeurs AMD Opteron dual-core 246 à 2GHz/1MB/400MHz et de 2GB de mémoire vive avec une interface GigaEthernet (pilote `tg3`). Ces machines physiques hébergent pour nos expériences une, deux ou quatre machines virtuelles *XEN*. Chaque machine virtuelle exécute un système d'exploitation GNU/Linux et a une partition disque dédiée (système de fichiers non-partagés). Le noyau Linux modifié pour *XEN* des domaines invités est la version `2.6.18`. Les machines physiques sont interconnectées directement via un commutateur Gigabit ethernet.

Afin de charger de manière contrôlée le processeur, nous avons développé un programme spécifique de simulation de charge utile, appellé le *softloader*. Il est constitué d'une boucle active effectuant des calculs en arithmétique entière, suivie d'une instruction de pause paramétrable (`nanosleep(long)`) qui nous permet de charger le processeur à une valeur déterminée (ex : 50%, 90%...). Pour générer et mesurer le trafic réseau nous avons utilisé l'outil de benchmark `iperf` [TIR] afin de générer des flux TCP depuis l'espace utilisateur des différentes machines virtuelles. Nous avons généralement utilisé le paramétrage par défaut de iperf. La taille de fenêtre de TCP n'a pas été modifiée directement mais via iperf. Pour superviser la charge des processeurs (ou plus exactement des *core*s) nous avons utilisé `sar`. Ainsi cet utilitaire nous permet d'obtenir les mesures de références (sans virtualisation). Nous avons utilisé une version modifiée par nos soins de `xenmon` (programmé en Python) pour mesurer cette charge sous *XEN*

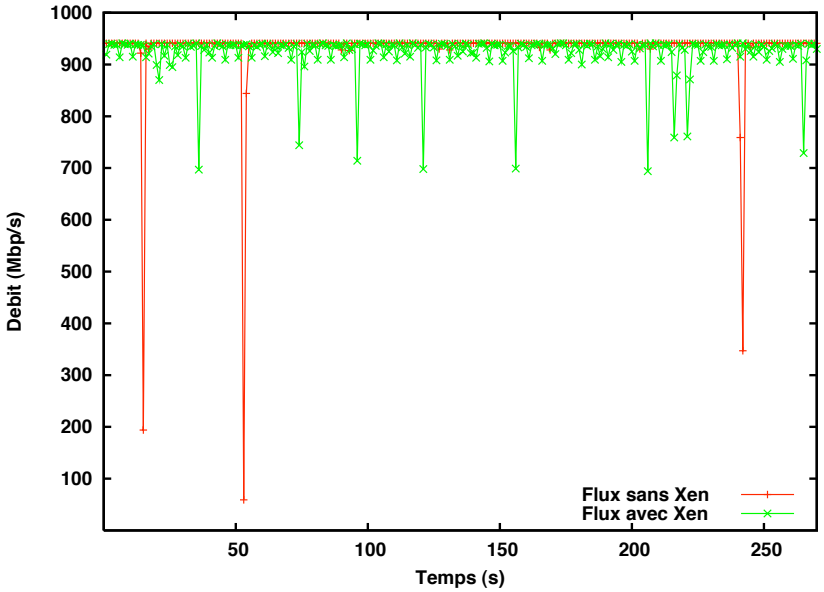


Figure 3. *Débit réseau (avec et sans XEN)*

4.2. *Résultat de l'expérience 1 : coût pour une machine virtuelle*

L'objectif de cette expérience est d'évaluer les performances de TCP dans le cas d'une machine utilisée uniquement pour la communication puis dans le cas d'une machine utilisée pour la communication et pour le calcul.

Lorsque la machine n'est sollicitée que pour communiquer, on observe que le débit réseau est très faiblement impacté par la présence de *XEN*. Il est de 938 Mbits/sec au lieu de 941 Mbits/sec maxi constaté sur cette configuration sans virtualisation). La figure 3 illustre ce modeste écart de débit, mais met aussi en évidence des chutes de débit sporadiques. La figure 4 montre le coût CPU brut de *XEN* (distance entre les courbes). On observe que le CPU n'est utilisé qu'à 30 % sans virtualisation. Le fait d'utiliser *XEN* entraîne un surcoût de 20 à 25% et les mesures mettent en évidence une instabilité de la charge (de l'ordre de 10%). Nous avons constaté un résultat comparable au niveau du serveur avec un coût CPU de base légèrement plus important (35%). Dans le cas d'une machine virtuelle exécutant simultanément une communication gigabit et un calcul saturant un cœur à 90 % nous observons que les performances réseau sont peu affectées La chute en débit est de quelques Mbps. Ces résultats encourageants sont néanmoins dus à la configuration bi-coeur des machines employées. La gestion des coeurs est faite par *XEN* et les traitements protocolaires ne sont pas systématiquement exécutés sur le même processeur.

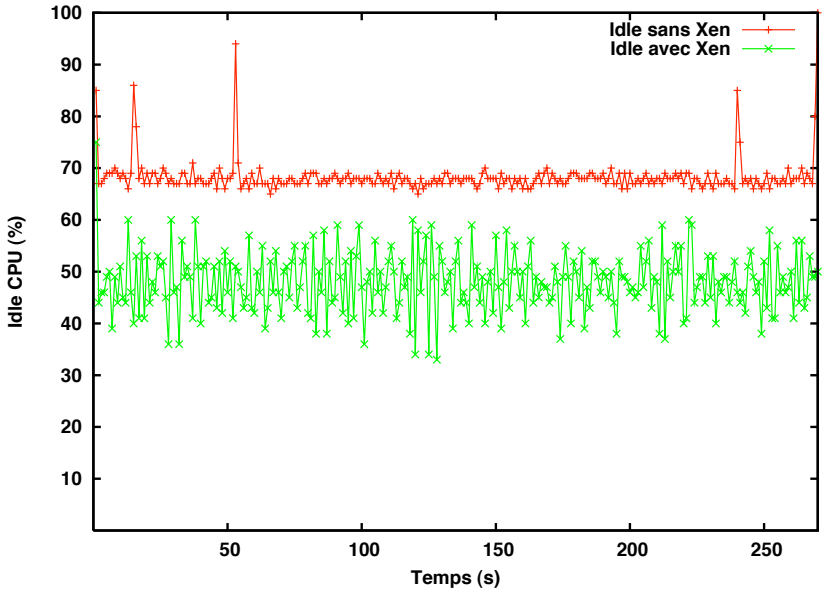


Figure 4. *Ressource CPU disponible coté client (avec et sans* XEN*).*

4.3. *Résultats de l'expérience 2 : taille des buffers*

Dans le cas de l'utilisation de liens gigabit, il est connu que le paramétrage TCP (TCP tuning) a une grande importance. Aujourd'hui les valeurs optimales sont connues et les systèmes d'exploitation activent généralement l'autotuning. Dans cette série d'expérience, la taille des fenêtres TCP a été laissée par défaut. Cette taille est automatiquement calculée et optimisée par le noyau. On note par exemple que cette taille de fenêtre n'est pas la même pour le domaine 0 (64 Ko) que pour les domaines invités ou virtuels (16 Ko) ce qui affecte les performances réseau : Communications Intra-machine
Domaine 0 vers Domaine virtuel (TCP Win 64Ko) : 610 Mbits/sec
Domaine virtuel vers Domaine 0 (TCP Win 16Ko) : 2.05 Gbits/sec

Par ailleurs, nous avons mené des expériences entre les domaine 0 et les domaines U entre 2 machines physiques distinctes.

Machine Serveur			
Machine Cliente		Domaine 0	Domaine U
	Domaine 0	941Mbits/sec	925Mbits/sec
	Domaine U	941Mbits/sec	930Mbits/sec

Le débit entre deux domaine 0 est comparable à celui que l'on pourrait obtenir avec une configuration sans *XEN*. Les débits entre les domaines 0 et les domaines virtuels sont moins bons à cause de la mauvaise sélection de la taille des fenêtres TCP. La communication entre les domaines virtuels et les domaines 0 reste très bonne et ce malgrè une taille de fenêtre TCP de 16Ko. Il faut noter que dans le cas de cette expérience la latence est très faible ($< 1ms$). La communication inter domaines virtuels (la plus commune) présente de bonnes performances avec un surcoût de 10%.

4.4. *Résultats de l'expérience 3 : Coût avec plusieurs machines virtuelles*

L'objectif de cette expérience est d'étudier le comportement dans le cas de plusieurs machines virtuelles s'exécutant sur la même machine physique. Nous avons considéré ici uniquement des configurations symétriques : N machines virtuelles clientes sur une machine physique communiquant avec N machines virtuelles clientes sur une autre machine physique serveur. Nous avons fait varier N selon les valeurs 1, 2 et 4. Nous donnons les courbes correspondant au cas de 4 machines virtuelles avec uniquement des communications (*i.e* sans charge).

	1flux 1VM		2 flux 2 VM			4 flux 4 VM				
	TCP1	Total	TCP1	TCP2	Total	TCP1	TCP2	TCP3	TCP4	Total
A vide (Mb/s)	940	940	500	410	910	350	215	155	115	835
En charge (Mb/s)	938	938	290	290	580	60	50	60	60	230

Figure 5. *Débits obtenus en fonction du nombre de machines virtuelles (en Mbits/s).*

La table 5 donne l'évolution des débits avec et sans charge dans le cas de 1, 2 et 4 machines virtuelles. On observe une réduction quasi linéaire du débit aggrégé ainsi qu'un accroissement de l'inéquité interflux dans le cas des processeurs à vide. En situation de charge, les débits s'écroulent avec l'augmentation du nombre de machines virtuelles (figures 6 et 7).

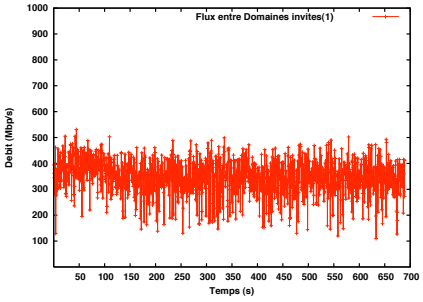


Figure 6. *Débit du flux TCP1 - cas 4 doms*

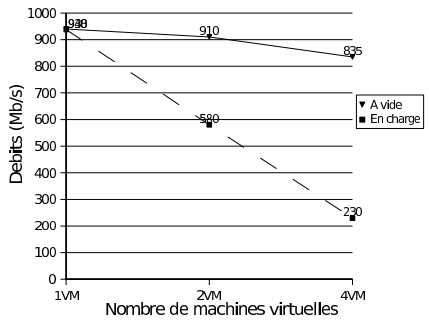


Figure 7. *Débits réseau aggrégés*

Les figures 8 et 9 explicite la répartition de la consommation des ressources dans les domaines 0 et U selon le nombre de machines virtuelles. On constate que cette consommation de ressources ne s'accroit pas linéairement. En particulier pour le domaine 0. En cas de charge, les communications semblent être fortement pénalisées au profit du calcul.

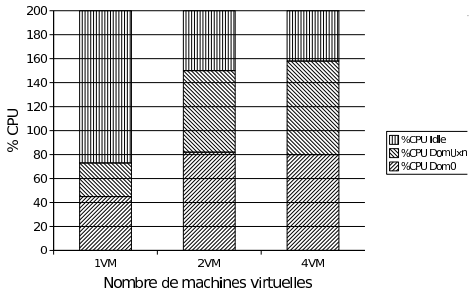


Figure 8. *Répartitions du CPU lors de transferts (à vide).*

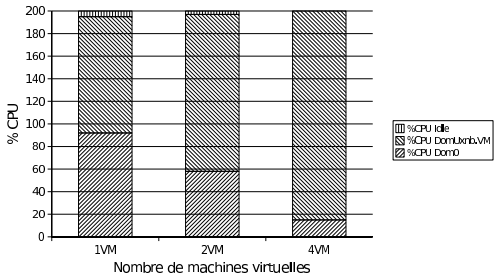


Figure 9. *Répartitions du CPU lors de transferts réseau (en charge).*

5. Travaux relatifs

La littérature dans le domaine de l'évaluation des performances réseau des systémes virtuels reste encore très restreinte. [MEN 07] montre les résultats du "profilage" de la machine virtuelle *XEN* pour déterminer où se situe les surcoûts. Ces résultats sont complémentaires aux notres même s'ils montrent des performances sous-optimales pour les connexions gigabit, étant donné que l'obtention du gigabit est dépendante de nombreux paramètres matériels. Il semble en effet que laconfiguration matérielle utilisée soit limitée. Dans [**?**] trois techniques intéressantes pour optimiser les performances réseaux de l'environnement virtuel *XEN* sont proposées. (1) les interfaces réseau virtuelles des domaines invités sont redéfinies pour profiter des fonctionnalités de haut-niveau disponibles sur les cartes réseau moderne (*offloading*). (2) l'implémentation du chemin de transfert des données entre le pilote du domaine invité et celui du Dom 0 sont optimisées. (3) le domaine invité a la possibilité d'accéder directement à la mémoire virtuelle (via des *superpages* ou des pages globales mappées).

6. Conclusions et travaux futurs

Les résultats obtenus montrent que le débit d'une connexion seule sur un lien Gigabit n'est quasiment pas affecté par la virtualisation même dans le cas d'une machine chargée à 90% (sur un cœur). Néanmoins, ce débit est obtenu au prix d'une consommation CPU supérieure et d'un ajustement des tailles de buffers. Lorsque l'on multiplie le nombre de machines virtuelles nos mesures montrent que le surcoût dû à la

virtualisation augmente linéairement avec le nombre de machines. On observe également un accroissement important de l'inéquité entre flux ce qui risque de poser des problèmes dans le cas de l'utilisation de la virtualisation dans les routeurs.

En termes de travaux futurs, nous nous proposons de mener une étude détaillée conjointe de l'ordonnancement des flux au niveau du domaine 0 et de l'ordonnancement des tâches dans *XEN*. Nous souhaitons aussi étudier les coûts lors de l'utilisation d'un routeur en domaine 0. Par ailleurs, les machines que nous avons utilisé pour nos expérimentations ne proposent pas le support de la virtualisation matérielle. Il serait interressant de refaire les mêmes mesures sur des machines offrant le support matériel de la virtualisation. Notre prochain objectif est d'évaluer le surcoût de la virtualisation sur le délai et la jigue. Enfin, à plus long terme nous souhaitons tirer avantage des cartes d'interfaces réseau programmables (équipés de *Network Processor*) en y déportant le switch/router virtuel ce qui devrait soulager le ou les processeurs physiques pour l'arbitrage des flux issus de domaines invités multiples.

7. Bibliographie

[BAR 03] BARHAM P., DRAGOVIC B., FRASER K., HAND S., HARRIS T., HO A., NEUGEBAUER R., PRATT I., WARFIELD A., « Xen and the art of virtualization », *SOSP '03 : Proceedings of the nineteenth ACM symposium on Operating systems principles*, ACM Press, 2003, p. 164–177.

[CAP 06] CAPPELLO F., DESPREZ F., DAYDE M., JEANNOT E., JEGOU Y., LANTERI S., MELAB N., NAMYST R., VICAT-BLANC PRIMET P., RICHARD O., CARON E., LEDUC J., MORNET G., « Grid5000 : a nation wide experimental grid testbed », *in International Journal on High Performance Computing Applications*, , 2006.

[GEL 03] GELINAS J., « Virtual Private Servers and Security Contexts », *http ://www.solucorp.qc.ca/miscprj/s_context.hc*, 2003.

[HOX 02] HOXER H.-J., BUCHACKER K., SIEH V., « Implementing a User Mode Linux with Minimal Changes from Original Kernel », *Proceedings of 9th International Linux System Technology Conference*, Cologne, Germany, September 2002.

[LES 96] LESLIE I. M., MCAULEY D., BLACK R., ROSCOE T., BARHAM P., D. EVERS R., FAIRBAIRNS, HYDEN E., « The design and implementation of an operating system to support distributed multimedia applications », *IEEE Journal on Selected Areas In Communications, 14(7) :12801297*, , 1996.

[MEN 07] MENON A., ZWAENEPOEL W., SANTOS J. R., TURNER Y., JANAKIRAMAN J., « Diagnosing Performance overheads in the Xen virtual machine environment », *Xen Summit 4, Yoktown, NY, USA*, , 2007.

[PRA 05] PRATT I., FRASER K., « Xen 3.0 and the Art of Virtualization », *Linux Symposium*, , 2005.

[QUM] QUMRANET, « KVM », `http://kvm.qumranet.com/kvmwiki`.

[TIR] TIRUMALA A., QIN F., DUGAN J., FERGUSON J., GIBBS K., « iperf : testing the limits of your network », `http://dast.nlanr.net/Projects/Iperf/`.

Tests & Sécurité

Automatisation de tunnels IPSec imbriqués pour établir des communications anonymes

Hervé AIACHE, Matteo LAURIANO, Corinne SIEUX et Cédric TAVERNIER

Thales Communications

160, bd de Valmy
92 704 Colombes Cedex
FRANCE
firstname.name}@fr.thalesgroup.com

RÉSUMÉ. *De nos jours, la sécurité et le respect de la vie privée sont devenues des challenges techniques critiques pour les courantes et futures générations de système de communication. Depuis les années 1980, beaucoup de systèmes définissent des solutions, principalement dérivées de l'approche bien connue des réseaux de MIX (développée par Chaum), visant à assurer l'anonymat des flux au niveau du réseau. Bien qu'efficaces, ces solutions souffrent d'un manque d'intégration d'approches IP standardisées et par conséquent, ne permettent pas une large adoption par le grand public. Ce papier propose une solution d'établissement de circuits anonymes, dérivée du très célèbre concept MIX, et bénéficiant du standard IPSec. Cette solution a été implémentée et expérimentée sur un vrai démonstrateur dans l'optique d'analyser les impacts de la solution sur la transmission de flux de bout en bout.*

ABSTRACT: *Nowadays, security and privacy are becoming two of the most critical issues for current and future generation of communications systems. Since the 80's, many efficient systems have been proposed to ensure flows anonymity, mainly derived from the so-called Chaum's Mix networks. However, these solutions suffer from a lack of integration with standardized IP approaches and therefore missed a wide adoption by the general public. This paper proposes an anonymous circuit establishment scheme derived from the powerful Mix networks concept and inheriting from the IPSec Framework. This solution has been implemented and experimented over a real testbed in view to analyze its impacts on multimedia flows end-to-end transmission.*

MOTS-CLÉS : : *Anonymous routing, Privacy, Traffic Flows Confidentiality, IPSec, Chaum's Mix*

1. Introduction

De nos jours, la sécurité et le secret de la vie privée sont devenus deux des plus importants problèmes des systèmes de communication présents et futurs. De nombreux fournisseurs de services demandent (et dans la plupart des cas collectent) des informations personnelles (identité, numéro de compte, mot de passe, lieu, préférences…) afin d'accéder et d'exploiter ces données pour le développement de services totalement adaptés à la personne (i.e. stockage numérique, services de location, ou accès au comptes bancaires. Des personnes malveillantes qui observent simplement le réseau, peuvent acquérir facilement des données sensibles et privées sur les usagers. Ces données peuvent être chiffrées, mais, la protection des informations privées requiert d'autres techniques qui permettent de masquer non seulement la source, la destination du trafic, le chemin utilisé et les données mais aussi le type de trafic que l'usager génère, et plus généralement leur comportement au sein du réseau.
Depuis les années 80, de nombreux systèmes efficaces (e.g. [4], [5], [3]), principalement dérivés des solutions basées sur le concept de Mix [4], permettant d'assurer l'anonymat des flux de communication ont été proposés pour permettre de résoudre ces aspects de protection de le vie privée. Cependant, elles souffrent d'un manque d'intégration basées sur des approches IP standards. Ceci peut expliquer le fait que ces solutions n'aient pas été largement acceptées du grand public.
Ainsi, ce papier présente une solution d'établissement de circuit anonymes dérivée du célèbre concept Mix, et héritant du cadre d'IPSec. Ce papier est structuré en trois parties : tout d'abord la section 2 présente pour la première fois une vue fonctionnelle, décrivant la plupart des solutions de routage anonymes existantes (e.g. Tarzan[5], Mix[4], OR[6], TOR[3], MorphMix[8]), la section 3 détaille la solution de circuit anonyme basée sur des tunnels IPSec imbriqués, la section 4 rapporte les résultats obtenus lors d'une expérimentation réelle, analysant les impacts des algorithmes de chiffrement sur des flux multimédia de bout en bout. La conclusion résume les différents aspects et présente les travaux futurs.

2. Vue fonctionnelle des approches de routage anonyme

Deux groupes de solution de routage anonyme existent : les architectures peer-to-peer et celles basées sur le concept « d'oignon ». Elles sont toutes les deux principalement basées sur une approche de nœud Mix, mais développées et distribuées dans un environnement dynamique pour la première catégorie (Tarzan, Crowds[7], MorphMix), et dans un environnement plus centralisé et fixe pour les autres (Chaum's Mix-net, mixMaster, Web-Mixes, TOR). L'analyse fonctionnelle des ces différentes approches de systèmes anonymes montre que la plupart des solutions de routage anonyme se basent sur quatre principaux composants et sur un bloc définissant des politiques de transmission.

Deux de ces composants (sections 2.1 et 2.2) sont plus liés aux aspects de routage (découverte de nœud, établissement de topologie, validation de confiance), et les trois autres (sections 2.3 à 2.5) sont plus liés au plan de retransmission (i.e. notion liées à la fonctionnalité de « forwarding »). Tous ces blocs fonctionnels sont illustrés à la figure 1 et sont décrits en détail dans la suite.

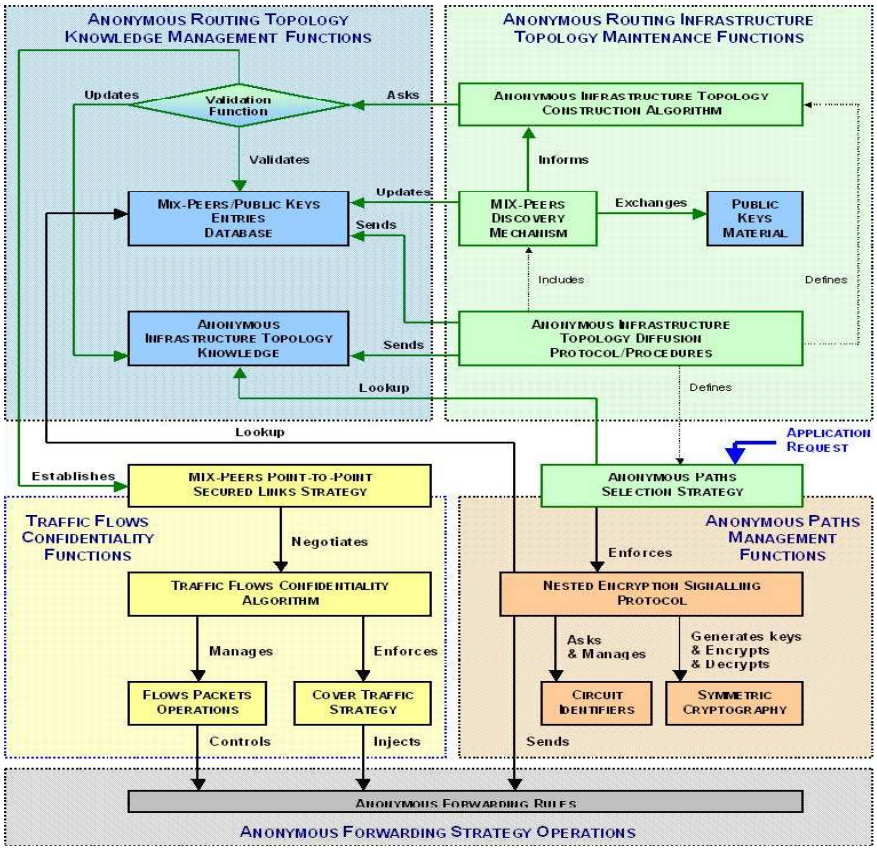


Figure 1. *Vue fonctionnelle des composants permettant une définition complète des solutions de routage anonyme*

2.1. *Maintenance de la topologie de routage anonyme (composant Anonymous Routing Infrastructure Topology Maintenance)*

Ce composant regroupe toutes les fonctions nécessaires à la gestion d'entrée et de sortie des nœuds Mix au sein de l'infrastructure de réseau anonyme. Remarquons que, en fonction du type de routage anonyme, ces opérations peuvent être opérées statiquement, par le biais de procédures (ou par un rafraîchissement basé sur un serveur central – e.g. TOR[3]) ou dynamiquement comme pour les solutions [2, 5], reposant sur des protocoles distribués spécifiques.

L'ensemble de fonctions implémentant la maintenance des topologies des infrastructures de routage anonyme définit dans la plupart des cas une construction de topologie particulière (e.g. CHORD, CAN ou Gossip pour P2P) afin de mieux contrôler l'infrastructure. Principalement associée aux fonctions de topologie, une stratégie particulière pour le choix du chemin anonyme est implémentée afin de choisir au mieux les nœuds Mix qui vont le composer.

2.2. Gestion de la connaissance de la topologie de routage anonyme (composant Anonymous Routing Topology Knowledge Management)

Ce composant intègre toutes les structures d'information (e.g. tables, bases de données) qui sont nécessaires pour savoir comment l'infrastructure est définie et quels sont les paramètres de chiffrement utilisés (principalement les clefs publiques).

Ces tables sont initiées et remplies durant la découverte des nœuds Mix et mises à jour régulièrement (et automatiquement) par le biais de protocoles de routages anonymes ou statiquement par le biais de procédures. Toutes ces informations vont être utilisées ensuite pour établir les circuits anonymes, que nous décrirons plus tard.

2.3. Composant TFC (Traffic Flows Confidentiality)

Ce composant permet d'auto-configurer les opérations permettant de masquer certaines caractéristiques de trafic des flux. Il repose principalement sur deux sous-fonctionnalités : la modification des paquets (i.e. ajout de données et opération de temps) et la gestion de « faux » paquets.

Il résout les attaques basées sur la corrélation des données analysées dans le temps. Dans la plupart des approches, ces problèmes peuvent être résolus par la définition d'algorithme particulier de TFC.

2.4. Gestion du chemin anonyme (Anonymous Paths Management)

Ce composant (une variante de l'approche basé sur le concept de Mix développé par Chaum) protége les identités de la source et/ou de la destination (i.e. les identifiants ou adresses des nœud Mix). Principalement, cette fonction basique permet d'installer un chemin anonyme basé sur une stratégie de sélection des nœuds Mix (spécifiés par le routage anonyme ou par le biais de procédures manuelles). Un message de signalisation basée sur le concept « d'oignon » (e.g. TOR[3], [6]) afin d'établir un chemin anonyme au sein de l'infrastructure anonyme en overlay. De plus, la gestion des chemins anonymes est articulée autour d'une stratégie particulière de chiffrement imbriqué (i.e. couches successives de chiffrement), qui est dans notre cas élaboré autour d'algorithmes de chiffrement symétrique.

2.5. Règles d'anonymat de la transmission (Anonymous Forwarding Rules)

Ces règles regroupent un ensemble de politiques nécessaire au niveau de l'envoi de la trame afin de garantir l'anonymat des flux de communication. Ceci est

nécessaire car les autres composants ne peuvent pas masquer toutes les informations nécessaires pour assurer proprement les règles de transmission. (i.e. les identifiants ou adresses des nœuds Mix).

3 Spécification et implémentation de la solution

Après quinze ans de recherche et de développements, les approches obtenues, bien qu'efficaces, n'ont pas été réellement intégrées au sein des infrastructures Internet. Ceci est principalement dû au fait que les solutions n'ont pas été élaborées autour des standards de sécurité basés sur IP[1]. Ainsi, pour combler ce manque, ce papier spécifie et définie une solution d'établissement de circuit anonymes chiffrés nativement inscrit au sein du standard IPSec, et une bibliothèque logicielle pour facilement l'implémenter. Cette solution répond aux attaques passives par analyse de trafic et ne permet pas de faire le lien entre la source et la destination (premier et dernier MIX). Dans notre approche fonctionnelle, cette solution correspond au composant de gestion de chemin anonyme (APSM : *Anonymous Path Signalling Manager*).

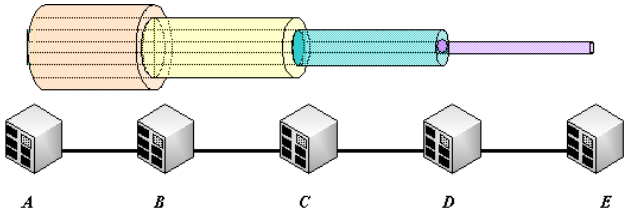


Figure 2. *Technique de tunnel imbriqués pour l'établissement d'un chemin anonyme*

Cette approche est complémentaire du standard RFC 4303 définit par l'IETF, qui propose un possible anonymat du trafic en dé-corrélant le trafic utile pour les attaques de temps (TFC). Un exemple illustrant le chemin anonyme établi, une fois les messages de signalisations échangés, est montré à la figure 2.

3.1. Spécification détaillée du logiciel

La gestion d'un chemin anonyme définie par le composant APSM peut être décomposée en deux opérations distinctes, dépendant de la position du nœud Mix considéré.

En effet, le nœud Mix peut initier l'établissement (ou le retrait d'un circuit anonyme (i.e. en étant le nœud source initiateur ou premier Mix du chemin), ou le nœud Mix peut être un nœud intermédiaire au sein du chemin anonyme sélectionné,

et dans ce cas, doit contribuer à l'établissement (ou au retrait local) de la connexion anonyme.

Ainsi, la distinction est importante afin de comprendre comment les composants logiciels (illustrés par la figure 3), vont agir différemment pour implémenter ces opérations .

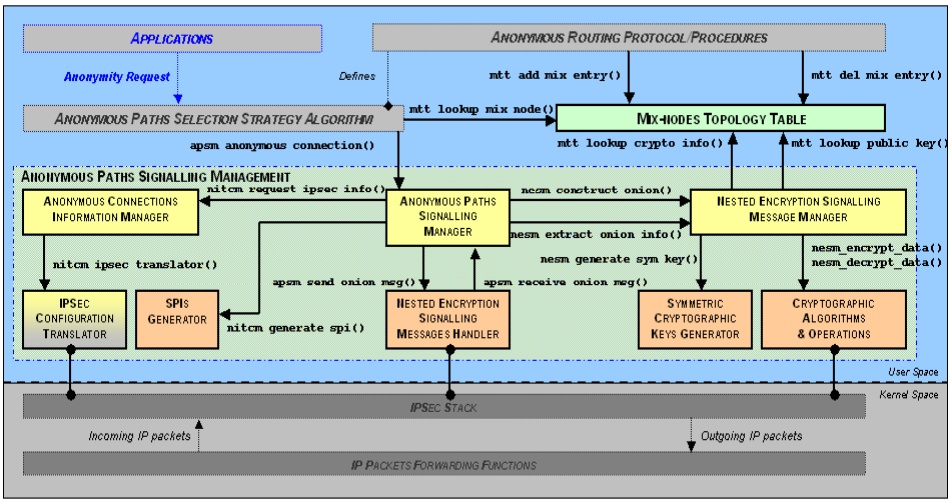


Figure 3. Détails des sous-composants de l'APSM

Dans le cas d'établissement du chemin anonyme, le nœud Mix initiateur va devoir générer les différents mécanismes de chiffrement pour établir le chemin. Dans le cas d'un nœud intermédiaire, le nœud reçoit un message de signalisation dérivé du concept « d'oignon », dont il va retirer une première couche, appliquer la configuration de tunnel IPSec correspondante et transmettre l'oignon de signalisation ainsi pelé, au nœud Mix suivant. Ces deux différentes phases sont expliquées dans les sections suivantes. L'utilisation de solution standardisée (IPSec) permet une intégration plus facile dans les infrastructures de réseau existantes.

3.2. Détails des opérations

3.2.1 *Opération d' établissement d'un chemin anonyme* (Nœud initiateur)

Après avoir reçu une demande d'établissement d'anonymat initié par une application, l'APSM (*Anonymous Paths Selection Manager*) demande l'ensemble des nœuds élus (déjà fait par le composant de topologie) pour l'établissement du chemin en appelant la fonction `apsm_anonymous_connection()`. Les adresses IP des nœuds Mix sélectionnés sont fournies comme paramètres d'entrée de cette fonction. Notons que l'ordre des adresses IP est important car le circuit anonyme va être établi en contactant un par un les nœuds correspondants dans l'ordre de départ indiqué (l'imbrication du chiffrement se fera aussi dans le même ordre).

Ensuite, le composant APSM demande la génération de plusieurs SPIs, dépendant du nombre de nœuds Mix impliqués en appelant la fonction `nitcm_generate_spi()` du sous-composant *SPIs generator*. Après cette opération, l'APSM demande la génération du message de signalisation imbriquée : ceci est fait par appel à la fonction `nesm_construct_onion()`du sous-composant *Nested Encryption Signalling Message Manager*.

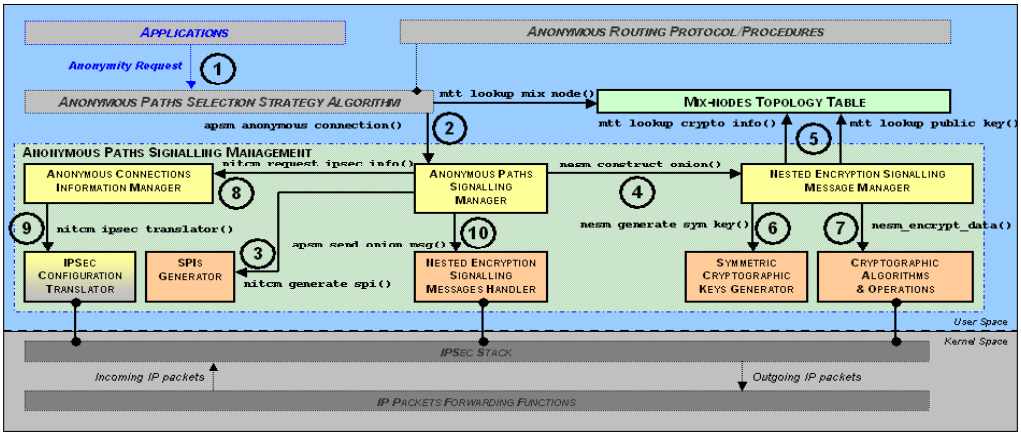


Figure 4. *Ordre d'appel des fonction de l'APSM pour l'établissement du chemin anonyme au sein du premier nœud Mix*

La liste des adresses IP des nœuds Mix et les SPIs précédemment générés sont donnés en paramètre en entrée de cette fonction.

Dans le but de générer le message imbriqué ("onion like"), le sous-composant *Nested Encryption Signalling Message Manager* récupère les clefs publiques (et les algorithmes de chiffrement associés) correspondant à la liste de nœuds Mix, en appelant la fonction mtt_lookup_public_key() (et la fonction mtt_lookup_crypto_info()). Ensuite, le sous-composant Nested Encryption Signalling Message Manager demande la génération de clefs symétriques en appelant la fonction nesm_generate_sym_key() du sous composant *Symmetric Cryptographic Key Generator*. Le message de signalisation, qui va être envoyé de noeud en nœud, avec une couche en moins à chaque nœud, est à présent construit, par appel à la fonction `nesm_encrypt_data()` du sous-composant *Cryptographic algorithms and operations.* L' APSM informe le sous-composant de gestion des circuits anonymes (*Anonymous Connections Information Manager)* par appel à la fonction `nitcm_request_ipsec_info()`. Les informations sont mémorisées localement, les politiques correspondant à la configuration de tunnels IPSec sont appliquées par appel à la fonction `nitcm_ipsec_translator()` du sous-composant IPSec configuration Translator.

Une fois les configurations appliquées, l'APSM envoie le message de signalisation au Mix suivant en appelant la fonction `apsm_send_onion_msg()`.

La figure 4 illustre l'ordre dans lequel les sous-composants opèrent entre eux.

3.2.2 Traitement d'un message de signalisation pour l'établissement du chemin anonyme (nœud intermédiaire)

Quand l'APSM reçoit un message depuis la signalisation, il arrive via le sous-composant *Nested Encryption Signalling Message Handler* au travers de la fonction `apsm_receive_onion_msg()`.

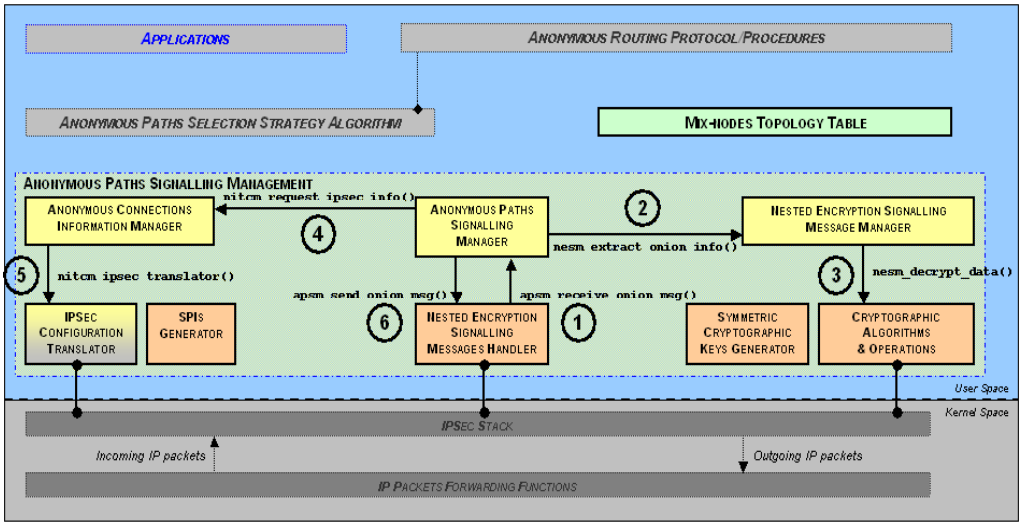


Figure 5. *Ordre d'appel des fonctions lors de l'établissement d'un chemin anonyme au sein d'un nœud Mix intermédiaire*

Une fois reçu, l'APSM retire une couche de chiffrement au message de signalisation en utilisant la fonction `nesm_extract_onion_info()` du sous-composant *Nested Encryption Signalling Message Manager*. Ensuite, ce sous-composant extrait les éléments de chiffrement au travers de la fonction `nesm_decrypt_data()` offerte par le sous-composant Cryptographic Algorithms and Operations en utilisant la clef privée du nœud Mix. Ensuite les informations concernant la configuration du tunnel IPSec (la clef symétrique, le SPIs et le prochain nœud du chemin anonyme) et celles concernant le message de signalisation imbriquée avec une couche en moins (ce message n'est pas déchiffrable car il est chiffré avec les clefs des nœuds suivants) sont séparées. Les informations concernant le tunnel sont remontées en paramètre de retour à la fonction `nesm_extract_onion_info()`. Le message pelé peut lui être directement envoyé au prochain Mix. L'APSM demande au sous-composant *Anonymous Connections Information Manager* d'appliquer les nouvelles demandes de circuit anonymes en appelant la fonction `nitcm_request_ipsec_info()`.

Comme précédemment, le sous-composant *Anonymous Connections Information Manager* crée et mémorise un nouvel état pour cette connexion anonyme. Il

demande au sous-composant *IPSec Configuration Manager* d'appliquer les différentes configurations ou commandes des tunnels IPSec en appelant la fonction `nitcm_ipsec_translator()`. Une fois informé de la mise en place du tunnel, l'APSM envoie le message de signalisation pelé d'une couche au prochain nœud Mix en appelant la fonction `apsm_send_onion_msg()`du sous-composant *Nested Encryption Signalling Message Handler*.

La figure 5 illustre l'ordre dans lequel les sous-composants opèrent entre eux.

4 Experimentations

Cette section décrit l'environnement dans lequel les mesures ont été réalisées. La plate-forme utilisée pour collecter les données est composé de 5 PCs, comme indiqué à la figure 6.

Les cinq PCs tournent sur un environnement Linux 2.6 et sont connectés en chaîne. Pour ce faire, quatre sous-réseaux sont définis. Trois d'entre eux supportent 100Mbps, le quatrième 10Mbps. Le but est de démontrer qu'il est possible de rendre anonyme et de sécuriser des communications pour du trafic temps réel.

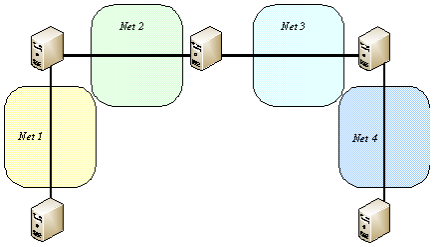


Figure 6. Plate-forme expérimentale

Dans cette section, les résultats des mesures sont présentés. Afin de simplifier la présentation, deux sous-sections sont définies. La première concerne les mesures avec ICMP, la deuxième celles réalisées avec UDP.

Cinq configurations ont été utilisées : sans tunnels, avec un, deux, trois, ou quatre tunnels. Deux différents algorithmes de chiffrement ont été utilisés : 3DES CBC et Rijndaël CBC avec ces trois longueurs de clefs différentes, i.e. 128, 192, et 256 bits. Pour chaque configuration, des mesures ont été faites pour des paquets dont la taille varie entre 64, 128, 256, 512, 1024, 2048 et 4096 octets.

4.1 Mesures avec ICMP

Le RTT (Round Trip Time) ou délai d'aller-retour dépendant de la taille des paquets pour un algorithme de chiffrement est présenté. Dans chaque figure, il y a

cinq courbes. Quatre d'entre-elles représentent les quatre configurations précédemment décrites (de 1 à 4 tunnels), la dernière représente le cas sans aucun tunnel (qui est le cas de référence).

Généralement, l'algorithme 3DES présente de mauvaise performances, comme indiqué par la figure 7. Si la taille des paquets est trop large, i.e. inférieur ou égal à 256 octets, il n'y a pas de grande différence dans le RTT, entre les configurations avec 1 ou 2 tunnels. Des différences significatives apparaissent pour des paquets de 1024 octets. Dans ce cas, la configuration avec 4 tunnels présente un RTT de 50% plus haut que le cas sans tunnel.

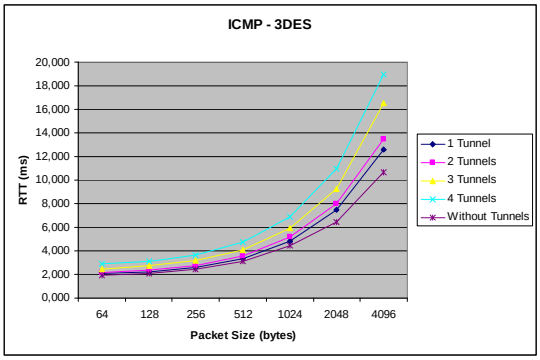


Figure 7. *RTT en fonction des paquets utilisant 3DES*

Les figures 8, 9 et 10 présente le RTT en utilisant l'algorithme de chiffrement de Rijndaël, avec respectivement des clés de 128, 192 et 256 bits. Il est important de souligner qu'entre les courbes il n'y a pas de grandes différences. Rijndaël 128 est plus rapide, mais les différences sont inférieures à 0,3 ms.

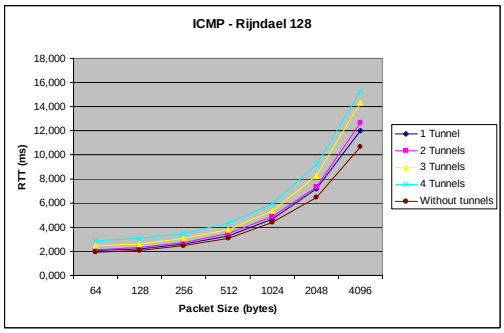


Figure 8. *RTT en fonction de la taille des paquets Rijndael 128.*

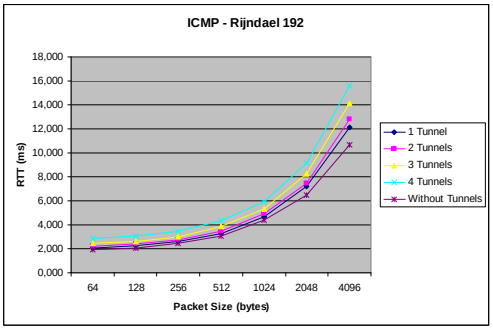


Figure 9. *RTT en fonction de la taille des paquets utilisant Rijndael 192.*

Le résultat est intéressant car cela signifie qu'il est possible d'incrémenter la sécurité de la transmission en utilisant une clé plus longue, avec un bon niveau de performance. Si la taille des paquets est trop large, i.e. égale ou supérieure à 1024 octets, de bonnes valeurs sont atteintes.

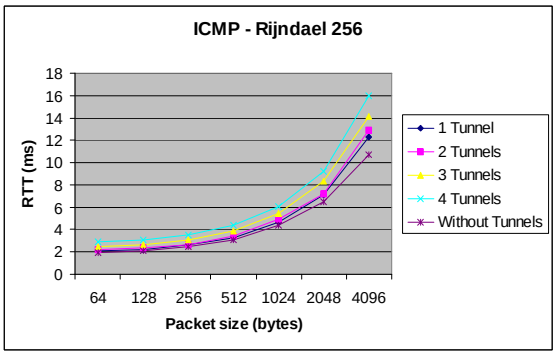


Figure 10. RTT en fonction de la taille des paquets utilisant Rijndael 256

4.2 Mesures avec UDP

Cette sous-section présente des mesures de gigue. Ces mesures sont évaluées pour un algorithme de chiffrement fixe, et de même que précédemment, on fait varier le nombre de tunnels. Pour chaque figure, différentes courbes sont disponibles, le paramètre étant la taille des paquets. La connaissance de la gigue est intéressante car la plus haute valeur ou variation de cette valeur peut déterminer des difficultés à transmettre des services temps-réel.

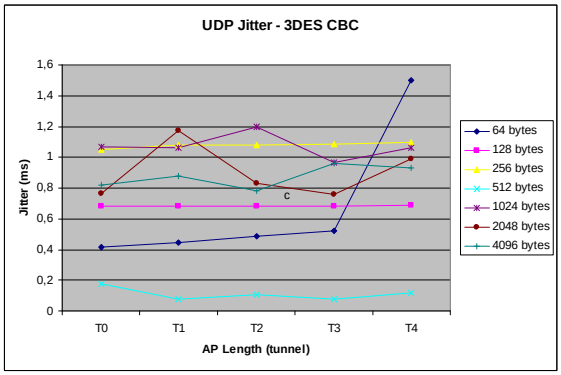


Figure 11. Gigue en fonction de la longueur du chemin anonyme en utilisant 3DES

La figure 11 montre la gigue mesurée quand l'algorithme 3DES est utilisé. 3DES n'est pas l'algorithme le plus mauvais, mais un comportement similaire est observé avec d'autres algorithmes. Les paquets de 128, 256 ou 512 octets ont le meilleur comportement car ils sont approximativement constants quand le nombre de tunnels change.

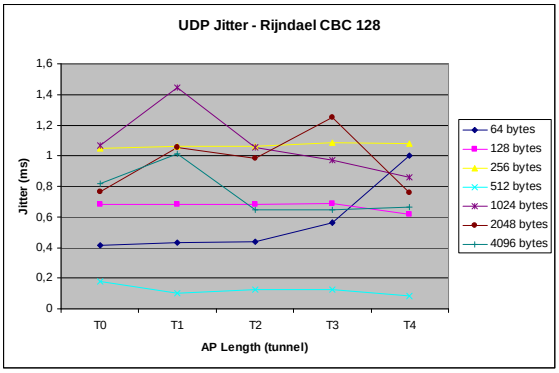


Figure 12. *Gigue en fonction de la longueur du chemin anonyme en utilisant Rijndael 128*

Le même comportement peut être observé sur les figures 12, 13 et 14. Les paquets de 512 octets ont la meilleure performance ; si 3DES ou Rijndaël 128 sont utilisés, le comportement est le plus régulier. Dans chaque cas, les paquets de 64 octets atteignent une haute valeur quand trois ou quatre tunnels sont utilisés. Les courbes pour des paquets de 1024, 2048 ou 4096 octets sont irrégulières. Généralement, tout d'abord les valeurs de gigues augmentent puis diminuent quand le nombre de tunnels augmentent.

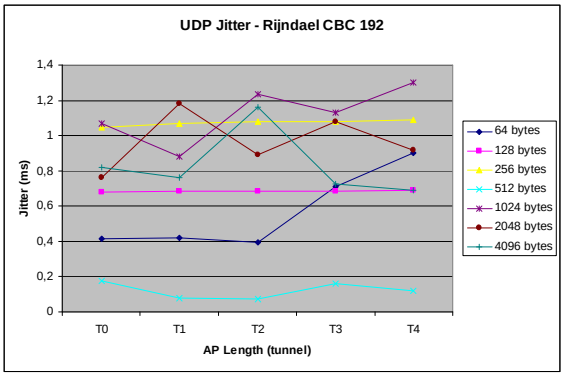


Figure 13. *Gigue en fonction de la longueur du chemin anonyme en utilisant Rijndael 192*

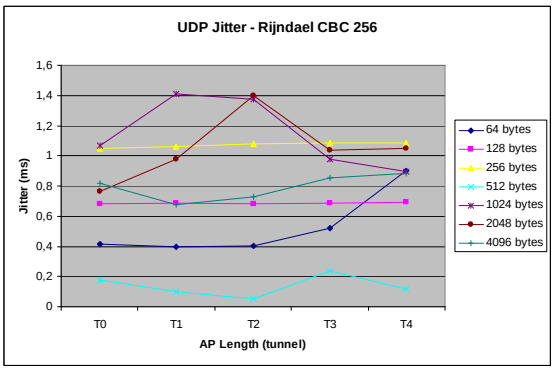


Figure 14. *Gigue en fonction de la longueur du chemin anonyme en utilisant Rijndael 256*

Les plus hautes valeurs sont atteintes pour des paquets de 1024 octets, quand Rijndaël est utilisé et quelque soit la longueur de la clé.

5. Conclusion et travaux futurs

Ce papier s'est intéressé et a présenté les principaux problèmes critiques que les réseaux IP doivent contrer aujourd'hui : la sécurité et la protection de la vie privée. Tout d'abord, ce papier a extrait un model fonctionnel qui s'applique à la majorité des solutions de routage anonymes présentées dans la littérature.

Nous avons ensuite présenté une solution basée sur des techniques de tunnels IPSec imbriqués afin d'assurer la non-traçabilité et non-observation des flux. La particularité de cette approche consiste principalement à la gestion de l'établissement de circuits anonymes dans un cadre standardisé. Un tel

environnement constitue un avantage certain pour un large déploiement ainsi que pour une adoption du grand public des solutions de routage anonyme.

De plus cette solution a été implémentée et testée sur une plate-forme expérimentale réelle afin de caractériser l'impact sur les flux multimédias. Nous avons pu observer que cette solution pouvait s'adapter à des applications nécessitant de vrais contraintes de temps réel .

Les travaux futurs consisteraient à inclure dans les performances et dans les améliorations déjà réalisées une implémentation de stratégie de découpage de message. De cette façon, les paquets IP envoyés par une source, seraient transmis à la même destination mais en utilisant des routes différentes afin de rendre plus difficiles les attaques passives. Dans ce cas, le nombre de tunnels IPSec et la longueur des circuits anonymes pourrait être diminués si de multiples routes sont utilisées. Ainsi, la performance globale serait améliorée, tout en conservant un niveau équivalent de protection et de sécurité. En d'autres mots, cette technique permettrait d'améliorer les performances de bout-en-bout en réduisant le nombre d'opérations de chiffrement et de déchiffrement. Les tests doivent aussi êtres réalisés en utilisant des applications audio et vidéo.

Nous souhaiterions remercier l'ensemble des partenaires du projet FP6 IST DISCREET, projet dans lequel nous avons défini cette solution et où nous testons en ce moment l'interaction avec le composant TFC identifié dans le model fonctionnel et le composant APSM. Pour plus d'information, merci de vous référer au site http://www.ist-discreet.org/.

7. Bibliographie et références

[1] S. Kent, "*IP Encapsulating Security Payload (ESP)",* IETF RFC 4303, December 2005.

[2] M. J. Freedman, R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer", *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002),* 2002.

[3] R. Dingledine, N. Mathewson, P. Syverson, "TOR: The Second-Generation Onion Router", *Proceedings of 13th Usenix Security Simposyum* , August 2004.

[4] D. Chaum, "*Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms",* Communications of the ACM, Vol. 24 Number 2, February 1981.

[5] M. J. Freedman, E. Sit, J. Cates, R. Morris, "*Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer"*, Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02), 2002.

[6] D. M. Goldschlag, M. G. Reed, P. F. Syverson, "*Hiding Routing Information"*, Workshop on Information Hiding, Cambridge, UK, May 1996.

[7] M. K. Reiter, A. D. Rubin, "*Crowds: Anonymity for web transactions"*, ACM TISSEC, June 1998.

[8] M. Rennhard, B. Plattner, "Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection", 2002.

Détection, classification et identification d'anomalies de trafic

Silvia Farraposo[1], **Philippe Owezarski**[2], **Edmundo Monteiro**[3]

[1] School of Technology and Management -Polytechnic Institute of Leiria
Alto-Vieiro, Morro do Lena, 2411-901 Leiria, Apartado 4163, Portugal
[2] *LAAS – CNRS / Université de Toulouse*
7 Avenue du Colonel Roche, 31077 Toulouse, cedex 4, France
3 DEI/CISUC - University of Coimbra
Pólo II – Pinhal de Marrocos 3030-290 Coimbra, Portugal

e-mail: silvia@estg.ipleiria.pt, owe@laas.fr, edmundo@dei.uc.pt

RÉSUMÉ. *Cet article présente un nouvel algorithme itératif – NADA – qui détecte, classifie et identifie les anomalies d'un trafic. Cet algorithme a pour objectif de fournir, en plus de ce que font d'autres algorithmes, toutes les informations requises pour stopper la propagation des anomalies, en les localisant dans le temps, en identifiant leur classes (e.g. attaque de déni de service, scan réseau, ou n'importe quel autre type d'anomalies), et en déterminant leurs attributs comme, par exemple, les adresses et ports sources et destinations impliqués. Pour cela, NADA repose sur une approche tomographique générique, multi-échelles, multi-critères et utilisant de multiples niveaux d'agrégation. De plus, NADA utilise un ensemble exhaustif de signatures d'anomalies qui ont été définies spécifiquement pour permettre de classifier ces anomalies. Ces signatures représentées sous forme graphique permettent une classification visuelle par les opérateurs réseaux. NADA a été validé en utilisant des traces de trafic contenant des anomalies connues et documentées comme celles collectées dans le cadre du projet MétroSec.*

ABSTRACT. *This paper deals with a new iterative Network Anomaly Detection Algorithm – NADA, which accomplishes the detection, classification and identification of traffic anomalies. Our approach goes one step further than others since it fully provides all information required to limit the extent of anomalies by locating them in time, identifying their classes (e.g., if it is a Denial of Service, a Network Scan, or other type of anomalies), and giving their features as, for instance, the source and destination addresses and ports being involved. For this purpose, NADA uses a generic multi-featured approach executed at different time scales and at different levels of IP aggregation. In addition, NADA uses an exhaustive set of anomaly signatures which have been specifically defined for anomaly classification purpose. These signatures, graphically represented, make possible a visual classification of anomalies by network operators. NADA has been validated using data traces containing documented anomalies as the one gathered in the MétroSec project.*

MOTS-CLÉS : *Anomalies de trafic, Identification, Signatures d'anomalies.*

KEYWORDS: *Traffic anomaly, Identification, Anomaly Signature.*

1. Motivations

Les anomalies de trafic en général et les attaques de dénis de service (DoS) en particulier sont un réel problème qui nuit à qualité de service dans les réseaux. C'est un des problèmes majeurs que les administrateurs voudraient pouvoir résoudre à la volée. Les anomalies font partie intégrante du trafic. Il est donc important mais aussi difficile de les détecter, de les classifier et de les identifier. Le traitement des anomalies détectées sera complètement différent suivant qu'elles sont légitimes – comme des foules subites – ou illégitimes – comme des attaques DoS.

La diversité des disfonctionnements du réseau a motivé la conception et le développement de NADA (Network Anomaly Detection Algorithm) dont les objectifs sont au nombre de trois :

– La détection des anomalies, i.e. déterminer si une anomalie est en train de se produire. En particulier, l'objectif sera de détecter toutes les anomalies, y compris les plus réduites. En effet, les attaques DoS sont de plus en plus souvent distribuées, chaque source de l'attaque ne générant que très peu de trafic, afin de rester invisible le plus longtemps possible. C'est en s'agrégeant massivement près de la victime que toutes ces composantes de l'attaque provoquent une dégradation brusque et importante du niveau de service fourni par le réseau et les serveurs qui y sont connectés : d'où l'intérêt de les détecter au plus tôt près de la source, i.e. lorsqu'elles ne représentent que peu de trafic ;

– La classification des anomalies, i.e. déterminer quel type d'anomalie est en train de se produire. Cela signifie déterminer si l'anomalie est légitime ou illégitime, et quel est son type précis (foule subite HTTP, scan réseau, attaque de SYN flooding, etc.) ;

– L'identification de l'anomalie, i.e. être capable de déterminer tous les paquets et flux de l'anomalie.

De plus, NADA est complètement générique. Il peut travailler sur n'importe quelle série temporelle produite à partir du trafic entrant (ou d'une trace de trafic enregistrée). Pour illustrer NADA, dans cet article, nous ne considèrerons que les 3 séries suivantes (mais NADA peut travailler sur un nombre bien plus important et un ensemble plus varié de séries temporelles si nécessaire) :

– le nombre de paquets par unité de temps;

– le nombre d'octets par unité de temps;

– le nombre de nouveaux flots par unité de temps.

Nous insistons fortement sur la nécessité de travailler sur plusieurs séries temporelles simultanément pour permettre des détections, classifications et identifications correctes des anomalies, et cela car chaque type d'anomalie agit différemment par exemple sur la liste des paramètres du trafic cités plus haut. De plus, comme nous avons conçu notre outil NADA à la demande d'opérateurs réseaux qui veulent pouvoir l'utiliser en toute maîtrise, il doit forcément considérer des caractéristiques représentative du réseau et de son

trafic, comme des octets, des paquets ou des flots, et cela en utilisant des statistiques simples. Ainsi, les séries temporelles sur le nombre de paquets SYN ou FIN pourraient être facilement utilisées par l'algorithme si elles sont nécessaires. Dans tous les cas, en utilisant uniquement des mathématiques simples, on souhaite faire de NADA un outil facilement et efficacement exploitable par des techniciens. Dans ce but, nous utiliseront une interface de communication simple, avec des représentations graphiques aussi souvent que possible.

Les capacités de détection d'anomalies de NADA à partir de l'analyse d'un trafic entrant permettent de mettre en évidence qu'une variation significative du trafic vient de se produire. Les capacités de classification permettent de signaler parmi un ensemble d'anomalies possibles de laquelle il s'agit. Les informations nécessaires à la classification sont produites par l'analyse simultanée des différentes séries temporelles, à différentes échelles de temps et avec différents niveaux d'agrégation. Enfin, les attributs d'identification dans le cas illustré dans cet article donnent des informations sur les entités impliquées dans l'anomalie, i.e. les adresses et ports source et destination. De plus, les distributions des attributs IP caractérisant un type d'anomalie ont une représentation graphique unique facilement identifiable. C'est cette signature graphique qui sera envoyée à l'opérateur réseau (technicien ou logiciel) pour qu'il lance la procédure corrective adéquate.

D'autres approches pour détecter les anomalies du trafic existent. Cependant, à notre connaissance, aucune d'entre elles ne permet de simultanément détecter, classifier et identifier les anomalies du trafic. La plupart des travaux dans ce domaine se concentrent sur la phase de détection. C'est le cas des premiers travaux comme ceux de Bardford et al. [1] et Krishnamurthy et al. [10]. D'autres étaient déjà plutôt orientés "anomalies", comme les travaux de Hussain et al. [7] qui ont propose une taxonomie des attaques DoS, les travaux de Jung et al. [8] qui ont étudié les foules subites, ou encore le système développé par Guo et al. [6] qui pouvait détecter et protéger les réseaux des scans réseau, par exemple. Une évolution majeure dans la lutte contre les anomalies a été l'introduction de capacités de classification, i.e. la capacité de donner un nom à l'anomalie qui s'est produite, notamment en utilisant les informations des couches TCP/IP. Des contributions importantes ont été celles de Kim et al. [9], Estan et al. [5] et Lakhina et al. [11]. A noter néanmoins que ces outils de détection ne fonctionnent qu'à partir de traces de trafic enregistrées. La classification d'anomalies n'en reste pas moins, encore aujourd'hui, un sujet trop peu abordé, et la conception d'une méthode de diagnostic des anomalies reste un problème ouvert. En particulier, aucune des approches décrites dans la littérature n'ont exploité complètement la richesse des attributs IP pour fournir des informations précises permettant d'identifier les responsables d'une anomalie. On peut pour cela mettre en cause les techniques mathématiques complexes (statistiques, traitement du signal) qui sont utilisées dans la plupart des cas pour la détection d'anomalies et qui rendent improbable un retour des espaces fréquentiel ou entropique vers des

attributs réseau facilement compréhensibles par les techniciens qui opèrent les réseaux.

La suite de cet article est organisée comme suit : la seconde partie donne les principes de la détection d'anomalies avec NADA. La partie 3 aborde l'aspect classification des anomalies. Pour cela, elle définit une signature pour les anomalies et montre des extraits de la base d'anomalies qui en découle. Puis elle décrit comment se fait la classification des anomalies. La partie 4, à la suite de la détection et de la classification des anomalies, décrit comment ces dernières sont alors trivialement identifiées. La partie 5 présente la validation de NADA à partir d'une base de traces de trafics réels dans lesquels des anomalies contrôlées ont été introduites. Enfin, la partie 6 conclut cet article en décrivant les améliorations qui sont en cours sur NADA.

2. L'algorithme de détection d'anomalies NADA

NADA a été défini comme un algorithme multi-échelles (il n'existe pas une échelle de temps unique qui permette de détecter toutes les anomalies – chacune agit avec ses propres caractéristiques temporelles), multi-critères (les anomalies n'affectent pas les attributs du trafic réseau de la même façon) et avec plusieurs niveaux d'agrégation (les différents types d'anomalies se perçoivent mieux à différents niveaux d'agrégation du trafic). Pour illustrer sur un cas concret les principes de conception de NADA, l'analyse du trafic entrant se fait, dans cet article, au niveau TCP/IP, en utilisant des informations sur les paquets et les flots, les flots étant définis selon Claffy et al [3]. La détection des anomalies dans NADA se fait en deux phases. La première a pour objectif de détecter les anomalies, alors que la seconde cible les flots anormaux pour les caractériser.

La première étape de NADA est un processus récursif qui commence par des mécanismes de détection peu coûteux (en temps) pour savoir si des variations significatives sur des séries temporelles de trafic fortement agrégé se produisent. Aux étapes successives, NADA augmente la quantité d'information à traiter en réduisant le niveau d'agrégation, et réalise alors des analyses plus fines.

NADA considère qu'une anomalie est responsable d'une variation sur au moins un des critères considérés, à au moins une échelle de temps et à un niveau d'agrégation donnés. Les échelles de temps Δ choisies pour analyser le trafic vont de quelques microsecondes jusqu'à plusieurs heures. Ces valeurs dépendent des objectifs fixés pour le temps de réponse de l'algorithme, mais aussi du type d'anomalie : certaines anomalies mineures quant à leur impact sur le réseau sont plus facilement détectables en considérant des fenêtres temporelles courtes. Les niveaux d'agrégation des adresses IP peuvent aller de la taille de préfixe /0 (tous les paquets sont considérés dans un flot unique pour l'analyse) jusqu'à /32 (chaque flot composé des paquets provenant ou allant vers une adresse IP unique sont analysés individuellement), en passant par les tailles de préfixes intermédiaires (pour lesquelles les flots contenant tous les

paquets venant ou à destination des adresses ayant un préfixe donné sont analysées individuellement). C'est en découpant ainsi à chaque étape le trafic en tranches tomographiques de plus en plus fines que l'on parvient à détecter les anomalies les plus massives et celles qui sont plus discrètes. Ainsi, dans le processus récursif, à chaque étape un nouvel ensemble de séries temporelles correspondant aux critères considérés est calculé pour une taille de fenêtre et une taille de préfixe donnée. A chaque étape ces séries sont analysées. De plus, pour chaque niveau d'agrégation /n des adresses, toutes les fenêtres temporelles sont étudiées. Et tous les préfixes d'adresses qui apparaissent dans le bloc de trafic considéré sont analysés. Dans les faits, et pour réduire le temps d'exécution en évitant des redondances entre des tailles de préfixes proches, on ne considère que quelques niveaux d'agrégation (e.g. /8, /16, /24, /32, ainsi que quelques valeurs intermédiaires si cela apparaît nécessaire au cours de l'une des itérations de l'algorithme).

Les variations sur les différentes séries temporelles représentant les différents critères (octets, paquets, flots) sont détectées en utilisant l'équation suivante : X est la série temporelle directement issue du trafic entrant. P est une série obtenue en faisant la différence entre toutes les valeurs consécutives de X deux à deux. Considérer P au lieu de X est un des points forts de cet algorithme : en effet, cela permet de ne détecter que les anomalies induisant une variation brusque et significative, et de ne pas être sensible aux variations normales du trafic. Ces variations importantes sur P ont été baptisées « deltoïdes » par Cormode et al. [4] qui les a aussi utilisés pour détecter les changements brusques dans le volume de trafic transitant sur un lien.

$$X = \{x_1, x_2, ..., x_n\}, x_i = \{\#\,paquets \,|\, \#\,octets \,|\, \#\,flots\}/\Delta$$

$$P = \{p_1, p_2, ..., p_{n-1}\}, p_i = x_{i+1} - x_i$$

$$\begin{cases} pi \geq E(p) + k\sigma \rightarrow anomalie \\ pi < E(p) + k\sigma \rightarrow normal \end{cases}$$

La moyenne E(p) et l'écart-type σ sont calculés pour chaque série temporelle et utilisés pour définir un seuil. Chaque dépassement de ce seuil, est interprété comme la présence d'une anomalie à cet instant. Suivant la valeur de *k*, ce filtrage peut être plus ou moins fin, selon l'intensité des anomalies que l'on cherche. Naturellement, des grandes valeurs de *k* conduisent à un plus grand nombre de faux négatifs, alors que de petites valeurs conduisent à un plus grand nombre de faux positifs.

Le résultat de cette première phase conduit à une liste de flots qui ont présenté une variation importante à un moment donné, et induisent donc potentiellement une anomalie.

3. Classification d'anomalies à partir de signatures

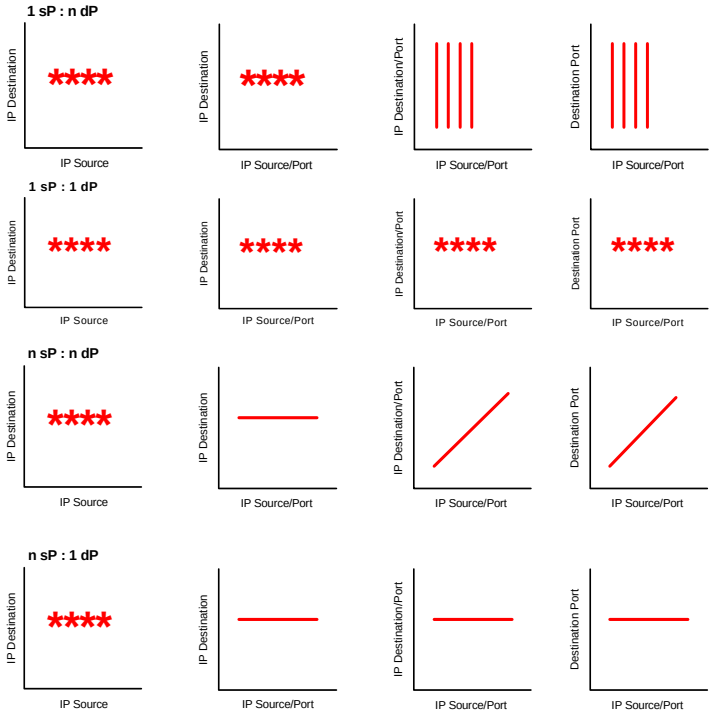


Figure 1. Exemples de s*ignatures théoriques d'attaques de DDoS de plusieurs sources IP vers 1 destination IP. Les différentes signatures représentent les cas pour lesquels un seul (resp. plusieurs) port(s) source est (resp. sont) utilisé(s) – 1 (resp. n) sP – et un (resp. plusieurs) port(s) destination est (resp. sont) ciblé(s) – 1 (resp. n) dP.*

La phase de détection précédente ayant permis d'isoler les caractéristiques des anomalies en fonction de critères relatifs aux couches TCP/IP, nous avons cherché à définir pour chaque type d'anomalies une signature unique qui leur est propre. En ne considérant que les nombres d'octets, de paquets et de flots, il est possible de définir un type d'anomalies à partir de 4 graphes bidimensionnels[1]. Sur chacun de ces graphes, chaque flot qui a été détecté comme comportant une anomalie est représenté par un point. Ces 4 graphes ont respectivement en abscisse et en ordonnée :

– l'adresse IP source et l'adresse IP destination ;

[1] Le choix de graphes bidimensionnels s'est imposé pour des raisons de lisibilité – avec des courbes en 3D les effets de perspectives étaient parfois trompeurs

– la concaténation de l'adresse et du port source, et l'adresse IP destination ;

– la concaténation de l'adresse et du port source, et la concaténation de l'adresse et du port destination ;

– la concaténation de l'adresse et du port source, et le port destination.

Ainsi, dans ces graphes, les différents types d'anomalies font apparaître des points ou des lignes : ce sont leurs signatures. A titre d'exemples, les figures 1 et 2 représentent respectivement des signatures pour des attaques DoS distribuées (DDoS) et des scans de réseaux. Prenons l'exemple d'une attaque DDoS de type *n*S, *n*P : *1*D, *n*P : plusieurs sources IP utilisant plusieurs Ports attaquent 1 destination IP sur plusieurs Ports (3ème signature sur la figure 1).

Sur le premier graphe, chaque point dénote les différentes sources (adresses IP) de l'attaque vers une seule et unique destination (au niveau IP). Lorsque l'on considère l'agrégation de l'adresse IP source avec le port source (second graphe), on voit pour une unique adresse destination une ligne droite qui montre que chaque source utilise différents ports lorsqu'il attaque la cible. Sur le troisième graphe, lorsqu'on agrège les numéros de port source et destination respectivement aux adresses IP source et destination il apparaît par la ligne droite en diagonale que plusieurs ports destination sont ciblés par l'attaque. Ceci est confirmé par le quatrième graphique par la ligne droite en diagonale montrant que plusieurs ports destination sont ciblés par l'attaque. Cette droite oblique met également en évidence une relation entre port source et port destination pour les paquets de l'attaque.

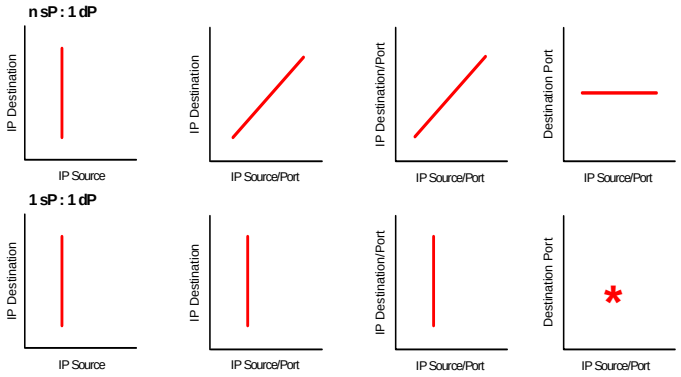


Figure 2. Exemples de s*ignatures théoriques pour des scans de réseaux (cf. figure 1 pour la légende)*

Il apparaît ainsi qu'avec ces représentations graphiques des signatures d'anomalies, il est très aisé d'analyser les caractéristiques des flots qui composent le trafic anormal et ainsi de classifier l'anomalie. Par manque de

place, les autres exemples donnés sur les figures 1 et 2 ne seront pas détaillés, de même que la base de signature complète ne peut pas être intégralement énumérée dans cet article.

Fort de cette base de signature caractérisant de manière unique un type d'anomalies, lorsqu'il détecte une anomalie, NADA la signale à l'opérateur et lui trace les 4 graphes de la signature. A partir de là, l'opérateur peut en quelques secondes déterminer quel type d'anomalie est en train de se produire, soit en analysant rapidement les 4 graphes, soit en se référant à la liste des signatures contenues dans la base et qui sont mises à sa disposition. A noter également que si une signature n'est pas encore référencée dans la base, parce qu'elle correspond à une attaque 0d (une attaque encore inconnue) par exemple, l'opérateur peut facilement comprendre son principe de fonctionnement en analysant les 4 graphes de la signature, en déduire la procédure corrective à appliquer, et enrichir la base de signatures.

4. Identification des anomalies

Etant donnés les informations continues dans la signature des anomalies détectées, l'identification des flots et paquets fautifs est immédiate. Toutefois, pour faciliter le travail de l'opérateur, celui-ci reçoit les informations requises en clair. Il n'a plus alors qu'à lancer les procédures correctives. Les données transmises en clair à l'opérateur sont la fenêtre temporelle dans laquelle l'anomalie est détectée, la liste des adresses IP et des numéros de port impliqués dans l'anomalie et la liste des adresses IP et des numéros de port ciblés

5. Validation

5.1. *Principe*

Valider NADA consiste à évaluer sa capacité à bien détecter les anomalies et à bien les classifier (l'identification étant triviale elle est de facto validée). La difficulté de la validation de détecteurs d'anomalies réside dans l'établissement des qualités de leurs performances statistiques. Les anomalies sont rarement annoncées a priori, de sorte qu'il est difficile de disposer d'un ensemble de traces, contenant des anomalies de types et caractéristiques connues, pouvant servir de base de données pour l'étalonnage des procédures de détection. Pour pallier cette difficulté, nous avons choisi de réaliser nous-mêmes un ensemble d'anomalies dont nous faisons varier les paramètres de manière contrôlée et reproductible. A partir de cette base de données que nous avons constituée dans le cadre du projet MétroSec [2], nous établissons les performances statistiques (probabilité de détections correctes versus probabilité de fausses alarmes) des procédures de détection proposées.

5.2. *La base de traces utilisées*

Cette base de traces de trafic contient plusieurs types d'anomalies, légitimes et illégitimes. Les anomalies ont été générées avec des intensités différentes pour évaluer la capacité de NADA à détecter des anomalies de faible intensité. A l'heure actuelle, notre base comporte 42 traces. Leurs caractéristiques et leur mode de production sont décrits dans [2]. Cette base contient notamment des traces d'**Attaques de DDoS** réalisées en utilisant différents types d'outils d'inondation (Iperf, Trinoo, TFN2k, etc.) qui ont permis de créer divers types d'anomalies de type « flooding ». Elle contient aussi des **Foules subites** sur un serveur Web.

5.3. *Validation des capacités de détection de NADA*

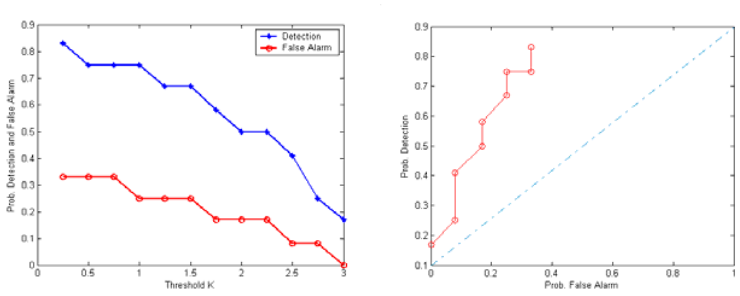


Figure 3. Performances statistiques de NADA pour la détection d'anomalies. A gauche, *Probabilité de détection PD = f(K) et probabilité de fausses alarmes PF = g(K). A droite, probabilité de détection PD en fonction de la probabilité de fausses alarmes.*

L'objectif de cette validation a consisté à confronter NADA aux traces de trafic avec anomalies contenues dans notre base. Toutefois, NADA possède un paramètre important qui influe sur les performances de l'algorithme : le paramètre de seuillage k. Pour cela, nous avons fait varier k dans l'intervalle]0, 3]. Pour chaque valeur de k, NADA a été évalué pour toutes les traces de notre base. La courbe à gauche de la figure 3 représente la probabilité de bien détecter une attaque et la probabilité de fausse alarme en fonction du paramètre k. Il montre en particulier que le taux de bonne détection dépasse toujours largement le taux de fausses alarmes. Il indique également que la valeur k = 2est une bonne valeur pour configurer NADA.

La courbe à droite de la figure 3 (obtenue par lecture sur le graphe de gauche de cette même figure) représente la probabilité de bonne détection en

fonction de la probabilité de fausse alarme. Le point idéal sur ce graphe est le coin haut à gauche pour lequel toutes les attaques sont détectées sans aucune fausse alarme. Le cas pire est la diagonale qui correspond à des résultats de détections aléatoires. Cette courbe montre que NADA est très efficace.

5.4. *Evaluation du principe de classification*

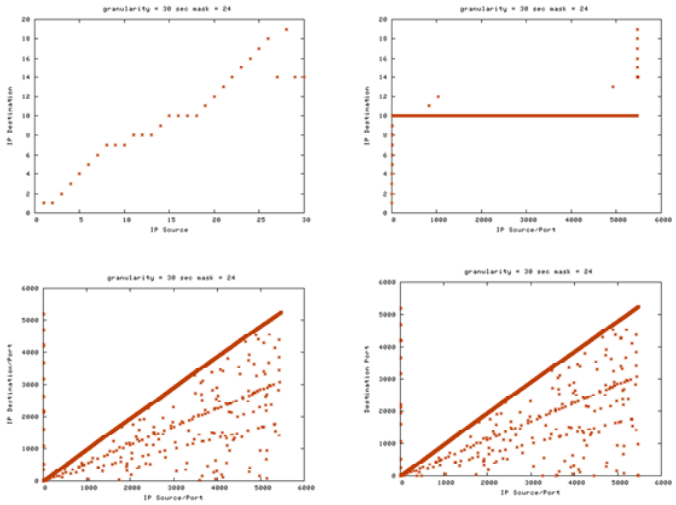


Figure 4. *Exemple d'attaque DoS par inondation*

A l'heure actuelle, NADA se limite à tracer les graphes de signatures lorsqu'il a détecté une anomalie. C'est donc l'opérateur qui est en charge de faire la classification par comparaison visuelle entre la signature de l'anomalie détectée par NADA et la base de signatures existante. De fait, cette partie ne présente pas une validation formelle, mais montre juste que les signatures connues (notamment celles représentées sur les figures 1 et 2) apparaissent clairement sur les signatures d'anomalies produites par NADA.

La figure 4 montre les 4 graphes produits par NADA lors de la détection d'une anomalie. Malgré le bruit sur les graphes (il existe des points qui ne sont pas sur les points ou les droites principaux de ces graphes), on voit clairement apparaître sur cette figure la signature d'une attaque de DDoS (la troisième de la figure 1 qui correspond à une attaque de plusieurs sources vers une cible et ce en utilisant des ports d'émission et de réception variés).

De même, malgré le bruit, il apparaît clairement sur la figure 5 la signature d'un scan réseau (cette signature est similaire à la première de la figure 2).

Ces deux exemples montrent clairement que la méthode de classification fonctionne. Les anomalies ont pu être facilement et rapidement reconnues visuellement. Cela a été le cas pour toutes les traces contenues dans notre base, mais également sur des traces inconnues qui se sont révélées contenir un certain nombre d'anomalies. Par manque de place, ce travail n'est pas rapporté dans cet article.

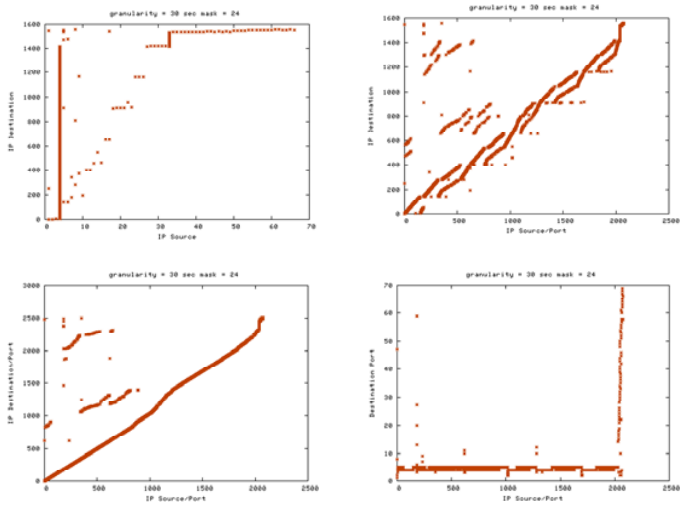


Figure 5. *Exemple de scan de réseaux*

6. Conclusion

Cet article a proposé une méthode de détection de classification et d'identification originale des anomalies du trafic. Ce travail bénéficie de nos connaissances sur les caractéristiques du trafic acquises depuis le début des années 2000 dans le cadre des projets Metropolis et MétroSec. Il en résulte un outil – NADA – très efficace qui repose sur une analyse multi-critère, multi-échelle et considérant divers niveaux d'agrégation du trafic entrant, et ce, à la volée. Une de nos contributions pour mener ce travail à bien a consisté à produire une base de traces de trafic contenant des anomalies connues et documentées, ainsi qu'une méthodologie de validation et d'évaluation statistique pour des outils de détection d'anomalies (ou d'attaques DoS).

Au niveau de la classification des anomalies, aujourd'hui, NADA laisse à l'opérateur la responsabilité de lire sur les 4 graphes de la signature le type d'anomalie qui est en train de se produire. Cette solution présente l'avantage de permettre d'analyser et de traiter ensuite de façon adéquate des anomalies inconnues. Toutefois, cela interdit de pouvoir lancer les procédures de correction automatiquement.

Aussi, pour transformer NADA en NAPA (pour assurer la **P**rotection du réseau en plus de la **D**étection des anomalies), nous allons concevoir un système de reconnaissance automatique du type d'anomalie. A priori, nous utiliserons pour cela des algorithmes de clustering permettant de mettre en évidence les points ou les lignes de fortes densités dans les graphes. Il sera alors possible de faire une comparaison automatique avec les signatures connues et de démarrer ensuite les corrections appropriées (éliminer les paquets d'une attaque par exemple, ou mieux gérer les ressources en cas de foule subite).

12. Bibliographie

[1] P. Barford, J. Kline, D. Plonka, and A. Ron. A Signal Analysis of Network Traffic Anomalies. In *Internet Measurement Workshop*, Marseille, November 2002.

[2] P. Borgnat, P. Abry, G. Dewaele, A scherrer, N. Larrieu, P. Owezarski, Y. Labit, L. Gallon, J. Aussibal, "Une caractérisation non gaussienne et à longue mémoire du trafic Internet et de ses anomalies : validation expérimentale et application à la détection d'attaque de DDoS", à paraître dans *les annales des télécoms*

[3] K. Claffy, H. Braun, and G. Polyzos. A Parameterizable Methodology for Internet Traffic Flow Profiling. In *Selected Areas in Communications – IEEE Journal*, vol. 13, p 1481-1494, October 1985.

[4] G. Cormode and S. Muthukrishnan. What's New: Finding Significant Differences in Network Data Streams. In *IEEE/ACM Transactions on Networking*, vol.13, n.6, pp. 1219-1232, 2005.

[5] C. Estan, S. Savage and G. Varghese. Automatically Inferring Patterns of Resource Consumption in Network Traffic. *In ACM SIGCOMM'03*, Karlsruhe – Germany, August 2003.

[6] Guo, X., Qian, D., Liu, M., Zhang, R., and Xu, B.: Detection and Protection Against Network Scanning: IEDP. In Proceedings of ICCNMC'01, Beijing (2001)

[7] Hussain, A., Heidemann, J., and Papadopoulos, C.: A Framework for Classifying Denial of Service Attacks. In ACM SIGCOMM, Germany (2003)

[8] Jung, J., and Krishnamurthy, B., and Rabinovich, M.: Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In WWW, Hawaii (2002)

[9] S. Kim, A. Reddy, and M. Vannucci. Detecting Traffic Anomalies through Aggregate Analysis of Packet Header Data. *In Networking 2004*, Athens – Greece, May 2004.

[10] B. Krishnamurthy, S. Sen, Y. Zhang and Y. Chen. Sketch-Based Change Detection: Methods, Evaluation, and Applications. In *Proceedings of IMC'03*, Miami – USA, October 2003.

[11] A. Lakhina, M. Crovella and C. Diot. Mining Anomalies Using Traffic Feature Distributions. *In ACM SIGCOMM*, Philadelphia – USA, August 2005.

Une approche de test de robustesse basée sur l'injection de fautes et le monitorage

Ana Cavalli* — Eliane Martins*, — Anderson Morais **— Bruno C.Moreira***

* *Groupe des Ecoles des Télécommunications/Institut National des Télécommunications – UMR CNRS SAMOVAR*
9, rue Charles Fourier, 91011 Évry cedex

Ana.Cavalli@int-edu.eu, bruno.chasco_moreira@hotmail.fr

** *Instituto de Computação- Universidade de Campinas*
Av. Albert Einstein 1251, 13083-970 Campinas SP Brazil
{eliane,anderson.morais}@ic.unicamp.br

RÉSUMÉ. Le test de robustesse d'un système a pour objectif de déterminer si ce dernier se comporte comme attendu en présence de conditions d'exécution erronées ou non prévues. Cet article présente une méthode pour le test de robustesse des protocoles, qui combine l'utilisation de techniques de monitorage et l'injection de fautes. Le monitorage est basé sur le test passif : les entrées et les sorties sont collectées pendant l'exécution du système étudié, puis analysées vis-à-vis de la spécification représentée par un modèle formel ou par un ensemble de propriétés. Dans cet article, l'injection de fautes, qui permet d'émuler les défaillances du réseau, est utilisée conjointement avec le monitorage pour tester la robustesse d'un protocole donné. Nous illustrons l'application de cette méthode sur un cas d'étude réel, le protocole WAP (Wireless Application Protocol) et, en particulier, sur le protocole WTP (Wireless Transaction Protocol). Afin de montrer la pertinence de la méthode proposée, les résultats obtenus de nos expérimentations sur l'injection de différents types de fautes sont présentés.

MOTS-CLÉS : *Test de protocoles, test de robustesse, test passif, injection des fautes.*

ABSTRACT. *Robustness testing has as main objective to determine if a system has the expected behaviour in the presence of faults. This paper describes a method for protocol robustness testing that combine monitoring and fault injection. Monitoring is based on passive testing: the inputs and outputs are collected during execution of the target system and further analyzed with respect to the specification represented either by a formal model or by a set of properties. In this paper, fault injection that allows to emulate communication failures, is used with monitoring to test the robustness of a given protocol. We illustrate the method application with a real world case study, the WAP (Wireless Application Protocol). Some experiments were performed, corresponding to the injection of a certain kinds of fault. The experimental results are presented in the paper.*

KEYWORDS: *protocol testing, robustness testing, passive testing, fault injection.*

1. Introduction

La robustesse d'un système peut être définie comme sa capacité de présenter un comportement *acceptable* en présence de fautes ou de conditions environnementales stressantes (Castanet et al., 2003).

Plusieurs approches pour le test de robustesse ont été proposées dans la littérature. Certaines s'inspirent du test de conformité incluant une phase de génération de tests basée sur le modèle formel représentant le comportement du système considéré. Dans ces approches, plusieurs types de modèles ont été utilisés (voir (Bochmann *et.al*, 1994) pour une présentation générale sur le sujet). La principale difficulté de ce type d'approches concerne la représentation formelle des fautes et leur contrôle (Khorchef *et al*., 2006). En effet, la définition d'un modèle formel pour la représentation des fautes est loin d'être triviale voire impossible (e.g, le temps entre deux entrées est plus court que le temps de réaction du système). De plus, il faut que ces fautes puissent être contrôlées pendant l'exécution par le *test harness* usuellement utilisé dans les tests de conformité.

Ces limitations nous ont amenés à choisir l'injection des fautes combinée avec le monitorage comme une technique mieux adaptée pour tester la robustesse d'un système. L'approche que nous avons définie procède en deux étapes. Dans la première étape, des fautes sont injectées dans le système, des traces d'exécution du système sont ensuite collectées, dans la seconde étape, pour être analysées en utilisant des techniques de monitorage afin de vérifier si l'implantation du système inclut bien des actions de recouvrement qui lui permettent d'être résilient aux fautes injectées.

L'injection de fautes consiste en l'introduction délibérée de fautes dans un système informatique afin d'observer son comportement. L'injection de fautes est cruciale pour les tests de robustesse, car elle accélère l'occurrence de fautes et d'erreurs qu'il serait difficile de recréer avec les techniques usuelles de test, ou qui nécessiterait trop de temps pour pouvoir les observer pendant l'exécution normale du système. Ainsi, à la place d'un *test harness* conventionnel, on utilise un injecteur des fautes qui peut s'insérer soit dans l'implantation sous test (IUT) elle-même, soit dans son contexte (bibliothèques de communication, couches inférieures ou même le matériel). Dans notre approche, l'injection des fautes est utilisée pour émuler les défaillances du réseau. Les fautes injectées ne sont pas choisies de manière aléatoire. Leur choix est basé sur l'observation des défaillances réelles ou sur des erreurs spécifiques que l'implantation est supposée tolérer. C'est cette dernière option que nous considérons dans ce travail. La plupart des méthodes de test de robustesse basées sur l'injection de fautes repose sur la définition suivante : « un système (une application) est considéré robuste si ce dernier ne s'arrête pas (crash ou hang) ». Une telle définition peut s'avérer assez restrictive. En effet, une défaillance n'arrête pas nécessairement le système, mais peut conduire l'implantation à des états erronés provocant une exécution incorrecte de ses fonctionnalités. Pour palier ce problème, notre approche utilise des techniques de monitorage basées sur des invariants pour

représenter des propriétés de robustesse. Ces propriétés sont soit fournies directement par les experts du réseau, soit déduites à partir de l'expression de besoins, soit encore à partir des modèles représentant le comportement du système étudié.

Les techniques de monitorage utilisées dans ce travail sont basées sur le test passif. Le test passif suppose l'observation du comportement de l'implantation sans interférer avec son exécution. La première étape consiste à collecter un corpus de traces. Une fois ce corpus obtenu, différents algorithmes sont appliqués pour vérifier que les traces collectées sont conformes à la spécification ou qu'elles vérifient certaines propriétés. Ces algorithmes sont des adaptations des algorithmes classiques de « pattern matching ».

Pour la mise en œuvre de notre approche nous utilisons deux outils : TestInv et Firmament. Le premier outil, développé à l'INT, permet d'automatiser toutes les étapes du test passif. Le second outil, Firmament, implémente un injecteur de fautes transparent pour l'utilisateur de l'application.

Pour illustrer la faisabilité et la pertinence de notre approche, le protocole WTP (Wireless Transaction Protocol) a été choisi comme cas d'étude. Ce protocole constitue l'un des composants imbriqués du protocole WAP (Wireless Application Protocol). WAP désigne une spécification globale ouverte, il permet à des utilisateurs mobiles d'accéder facilement, et de manière très rapide à l'information et aux services de l'Internet via des dispositifs mobiles. Au total, cinq expériences ont été réalisées. Chaque expérience est composée de plusieurs exécutions. Une exécution correspond à l'injection d'une faute. Dans cet article, nous détaillons les résultats de nos expériences sur le protocole WTP qui nous ont permis de détecter une défaillance dans ce dernier. Les expériences constituent à elles-seules une contribution originale de cet article, car ce type d'étude n'a pas été mené auparavant.

L'approche que nous proposons est originale car jusqu'à présent aucun travail de recherche dans le domaine n'a combiné ces deux techniques. De plus, elle permet de vérifier qu'un système exécute des actions de recouvrement en présence de fautes sans même avoir accès direct à son implantation. Ainsi, des couches protocolaires par exemple peuvent être testées bien que le testeur ne puisse pas envoyer directement les messages à la couche sous test.

L'article est organisé comme suit. La section suivante présente l'approche que nous proposons basée sur le monitorage et les techniques d'injection de fautes. La section 3 décrit les caractéristiques des deux outils, TestInv et Firmament, ainsi que les détails de leurs implantations. La section 4 décrit brièvement le cas d'étude: le protocole WAP. La mise en place de l'expérimentation de notre approche ainsi que les résultats obtenus sur le cas d'étude sont présentés dans la section 5. Nous terminons par une conclusion et des perspectives potentielles à notre travail.

2. L'approche proposée pour le test de robustesse

Cette section présente l'approche que nous proposons pour le test de robustesse. Nous commençons par décrire les types de fautes que nous considérons ainsi que les invariants utilisés pour le monitorage. Enfin, une vue générale des principales étapes de l'approche sont présentées.

2.1 Injection des fautes : types de fautes

Les types de fautes que nous considérons dans cet article sont basés sur les modes de défaillance suivants (Hadzilacos *et al*., 1994):

- défaillances d'omission : ce type de défaillances peut être émulé par l'interception soit de tous les messages provenant d'un hôte spécifique soit uniquement des messages envoyés ou reçus par un hôte.
- défaillances arbitraires : les défaillances de ce type, appelées aussi défaillances Byzantines, sont émulées par la corruption des messages reçus par l'IUT. Pour injecter les fautes Byzantines, il faut déterminer quels champs du message seront corrompus et avec quelles valeurs. Pour cela, on a utilisé l'approche proposée par Ballista (Kropp *et al*., 1998), où des valeurs spécifiques aux types des données sont choisies. Ainsi par exemple, pour des champs de type entier, des valeurs tels que -1, 0 et 1 peuvent être utilisées. Ceci permet la prise en compte des fautes de communication, mais également celles d'interface qui sont très utilisées pour tester la robustesse de fonctions dont les interfaces sont bien définies.
- défaillances temporelles : les défaillances de ce type peuvent être émulées par des délais de livraison de messages plus longs (ou plus courts) que prévus.

2.2 Le test passif

Les méthodes formelles pour le test peuvent être classées en deux grandes approches : actives et passives. Dans les approches actives, le testeur possède la capacité de stimuler l'IUT et de vérifier si la réponse obtenue pour chaque stimulus est en accord avec la spécification du système. Dans les approches passives, le testeur n'a pas besoin d'interagir avec l'IUT ; son rôle se limite à l'observation (ou monitorage) de l'exécution de l'IUT sans aucune interférence. Par conséquent, le testeur a besoin de points d'observation (PO) à partir desquels il peut collecter les messages échangés entre deux entités protocolaires. Dans ce cas, le testeur a deux rôles principaux : (i) observer, collecter et filtrer l'information échangée entre deux entités protocolaires. Le testeur est donc similaire à un analyseur (ou renifleur) de paquets, il intercepte et enregistre les messages passant sur la totalité du réseau ou

une partie de ce dernier; (ii) analyser les informations collectées (traces d'exécution) pour déterminer si elles sont conformes vis-à-vis de la spécification.

L'approche du test passif proposée dans cet article est basée sur l'analyse d'invariants. Les invariants sont des propriétés que la spécification du système doit satisfaire. Par conséquent, l'IUT devrait également les remplir pour être conforme à la spécification. Les invariants peuvent être extraits de la description informelle (ou textuelle) du protocole, ou de la spécification formelle de ce dernier. Dans le cadre de nos travaux de recherche, nous utilisons les machines d'états finis (FSM) pour la représentation formelle d'un protocole.

Définition d'invariant

Invariant simple

Soit *I* (resp. *O*) un ensemble de symboles d'entrée (resp. de sortie). Les invariants d'un système sont exprimés comme une succession de symboles d'entrée et de sortie. L'invariant (i_1/o_1, ..., i_{n-1}/o_{n-1}, i_n/O) est interprété comme suit: «chaque fois qu'on observe sur l'IUT la trace (i_1/o_1, ..., i_{n-1}/o_{n-1}) et qu'on observe l'entrée i_n alors on obtient une sortie qui appartienne à *O* ». Les invariants de ce type sont appelés des invariants *simples*. Les symboles "?" et "*" peuvent également être utilisés pour la définition d'invariants plus élaborés. Le symbole "?" est utilisé de la même manière que pour le « pattern matching », c'est-à-dire, il peut remplacer n'importe quel symbole. Cependant, la signification du symbole "*" a été légèrement modifiée. Par exemple, la signification de l'invariant (*i/o,*, i'/O)* est la suivante : si on détecte la transition *i/o* dans la trace, alors la première occurrence de *i'* est suivie par une sortie appartenant à *O*. Le symbole "*" remplace donc n'importe quelle séquence de symboles ne contenant pas l'entrée *i'*.

Invariant d'obligation

Bien que les invariants simples permettent la spécification de nombreuses propriétés des systèmes informatiques représentés par des FSM, ils ne sont pas toujours suffisants pour exprimer toutes les propriétés souhaitées. Dans le domaine des protocoles par exemple, on aimerait spécifier que l'occurrence d'un événement *e* est toujours précédée d'une séquence particulière d'événements. C'est pour cette raison que nous définissons un type supplémentaire d'invariant dit invariant d'*obligation*. Ainsi par exemple, l'invariant (1) signifie : si une sortie appartenant à *O* est observée, alors l'entrée *i* suivie de la sortie *o* a été observée auparavant.

$$i/o,*,i'/\overline{O} \qquad (1)$$

Dans le cadre de ce travail, les invariants sont représentés en utilisant l'EBNF (Extended Backus-Naur Form). Des exemples d'une telle représentation sont donnés à la Section 5.

Les invariants doivent être validés vis-à-vis de la spécification. La validation des invariants est nécessaire dans le contexte de notre approche car ces derniers sont fournis par l'équipe de test et peuvent donc être faux. En effet, ils ne sont pas

dérivés à partir d'un modèle formel. L'avantage de l'obtention des invariants par l'équipe de test est de se restreindre aux propriétés les plus pertinentes ou critiques.

2.3 Vue générale de l'approche proposée

Les différentes étapes de l'approche proposée sont décrites ci-dessous :

1ère étape : injection de fautes. Nous utilisons une approche basée sur l'insertion de code au niveau réseau qui peut affecter les messages des couches supérieures à travers l'utilisation de la capture et le filtrage de paquets. Nous posons comme hypothèse que ces fautes peuvent être provoquées par des défaillances dans le réseau ou dans l'entité paire.

2ème étape : *définition d'une architecture de test*. Mise en place de points d'observation (PO) qui seront utilisés pour la collecte de traces.

3ème étape : *application des techniques de monitorage*. Ceci implique :

- la collecte de traces d'une part, et d'autre part leur formatage pour les rendre directement utilisables par des techniques de monitorage;

- l'application des algorithmes d'analyse des invariants afin de déterminer si l'IUT a mis en place les actions de recouvrement des fautes injectées. Dans cette étape un verdict est donné suivant le succès ou l'échec de la validité des invariants.

3. Mise en œuvre de l'approche proposée

Dans cette section, nous présentons la mise en œuvre de notre approche par l'utilisation conjointe des deux outils TestInv et Firmament (voir Figure 1).

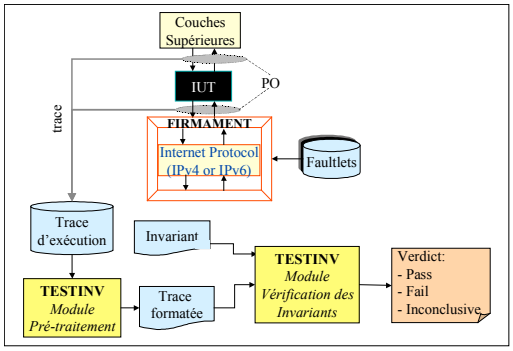


Figure 1. *Utilisation conjointe de TestInv et Firmament pour le test de robustesse.*

3.1. *TestInv*

L'outil TestInv (Bayse et al., 2005) est l'outil qui implante l'approche de test passif présentée dans la section 2.2. La figure montre deux des modules de TestInv. Le module *Pré-traitement* se charge du traitement des fichiers log contenant les corpus des traces observées. Le fichier est filtré et analysé grammaticalement afin d'obtenir une trace convenable, c'est-à-dire, une trace incluant l'information concernant les noms des primitives des entrées et sorties, ainsi que les données pertinentes aux messages échangés (l'adresse source, l'adresse de destination, l'url accédé, etc.). Ce module inclut une interface qui permet à l'utilisateur de paramétrer le format des traces à collecter. Ainsi, il n'est pas nécessaire de modifier le module *code* pour traiter les fichiers log qui proviendraient de différentes implantations, et par conséquent de différents formats. Enfin, le module *Vérification des Invaraints* détermine si les traces collectées satisfont bien la liste d'invariants donnée en entrée. L'outil a été complètement implémenté en JAVA (spécification J2SE 1.4.0).

3.2. *Firmament*

Firmament est une évolution de l'outil ComFIRM (COMmunication Fault Injector through operating system Resources Modification) (Drebes et al., 2005) lequel explorait l'extension du kernel Linux afin de développer un injecteur de fautes de communication transparent à l'utilisateur de l'application. Firmament permet la spécification de scénarios de fautes (*faultlet*). Un *faultlet* est exécuté sur chaque paquet qui traverse le flux de communication ; il a la capacité d'inspecter et de modifier les contenus d'un paquet, l'écarter, le dupliquer ou le retarder. Un *faultlet* est exécuté sur la machine virtuelle FIRMVM, qui est un module connecté au sous-système de communication à travers l'interface Netfilter. FIRMVM est responsable d'interpréter et de traiter les commandes qui spécifient les scénarios incluant des fautes.

4. L'implantation cible

Nous avons appliqué notre approche de test de robustesse au WAP (Wireless Application Protocol) (OMA 2001). Le WAP a pour but de permettre à des terminaux mobiles, tels que téléphones mobiles, PDAs (Personal Digital Assistant), ou autres, d'accéder à l'Internet.

Le WAP est structuré en sept couches successives. Nous présentons ici uniquement la couche WTP qui est la cible des tests réalisés. La couche WTP offre des services de transactions à sa couche supérieure (WSP) lorsque le mode connecté est sollicité. Cette couche utilise un service de datagramme fourni par la couche inférieure (WDP ou UDP). WTP offre des services du type requête-réponse. Le processus qu'initie une transaction est l'entité désignée comme *Initiator*, et la

réception de celle-ci est faite par l'entité *Responder*. La communication entre ces deux entités peut s'effectuer suivant trois niveaux différents de classe (0, 1 et 2). Ces niveaux de classes conditionnent le nombre de messages (en termes d'acquittement et de résultat) échangés entre les entités. Plus le niveau de la classe est élevé, plus le service est fiable.

Les interactions observables en entrée ou en sortie de la couche WTP pour un service de classe 2 sont : Invoke (pour initier une transaction), Result (pour retourner un résultat) et Abort (pour avorter une transaction). Ces interactions peuvent être des primitives de services échangées entre le WTP et le WSP, ou alors des PDUs (Protocol Data Units) transmises entre le *Initiator* et le *Responder*. Les PDUs correspondent aux interactions présentées plus haut, auxquelles on ajoute le PDU Ack, utilisée par les deux entités (émise par l'une ou par l'autre selon le niveau de classe adopté) pour acquitter un message reçu.

Une primitive peut s'écrire sous la forme suivante : TR-<nom>.<type>(<paramètres>), où : TR désigne la couche supérieure (WSP) ; <nom> correspond aux interactions présentées précédemment ; <type> peut prendre les valeurs suivantes : req (indique une demande de service faite par WSP), ind (informe la couche supérieure qu'une demande de service a été faite par l'entité distante), res (accuse la réception d'une primitive de type indication à sa couche inférieure), cnf (signale que la demande a été effectuée avec succès). Ces paramètres varient selon les types de primitives.

5. Les expériences réalisées

Cette section présente les expériences et résultats obtenus de l'application de notre approche sur le protocole WAP. À cet effet, la passerelle (ou *gateway*) WAP a été choisie comme cas d'étude pour réaliser l'interface entre le terminal mobile, fonctionnant sur un réseau sans fil, et le serveur. Dans ces conditions, la passerelle est exposée au modèle de fautes de communication mentionné en section 2.1. Par conséquent, sa robustesse par rapport à ce type de fautes doit être testée.

5.1. L'architecture des tests

La Figure 2 présente l'architecture mise en place pour la réalisation des tests de robustesse. Elle est composée de la passerelle WAP, mise en œuvre par un logiciel *open source*, Kannel[1], version 1.4.1. Cette passerelle a été utilisée dans le cadre de travaux précédents (Bayse et al 2005) , elle contient donc les points d'observation (PO) à travers lesquelles les informations, nécessaires pour la validation des invariants vis-à-vis de l'IUT, sont capturées dans des fichiers *log*. Chaque fichier contient les traces locales d'exécution d'une couche particulière du WAP.

[1] http://www.kannel.org/

Dans notre cas, le rôle du client (terminal mobile) est réalisé à travers un simulateur de téléphone WAP. Il s'agit du *Nokia Mobile Browser Simulator*[2] *4.0* (NMB4.0). C'est un logiciel qui est capable de simuler un téléphone WAP et qui dispose d'une interface graphique représentant un terminal mobile. L'interface fournie au client est un *browser* qui lui permet de demander à un serveur des pages en format WML. L'injecteur de fautes est situé dans la même machine que Kannel. Il capte tous les messages en entrée et en sortie comme l'indiquent les flèches différenciées (↑↓).

5.2. Description des tests

L'objectif de l'approche est de révéler la présence de fautes de robustesse dans l'implantation d'un protocole. Comme ces fautes internes peuvent être révélées en présence des fautes externes injectées, on a réalisé cinq expériences, composées de plusieurs *exécutions* (*run*), où chacune correspond à l'injection d'un type de fautes. Il est important de signaler qu'à chaque nouvelle expérience, on redémarre l'environnement de tests et l'IUT pour s'assurer que les éventuelles erreurs activées précédemment n'affectent plus son exécution.

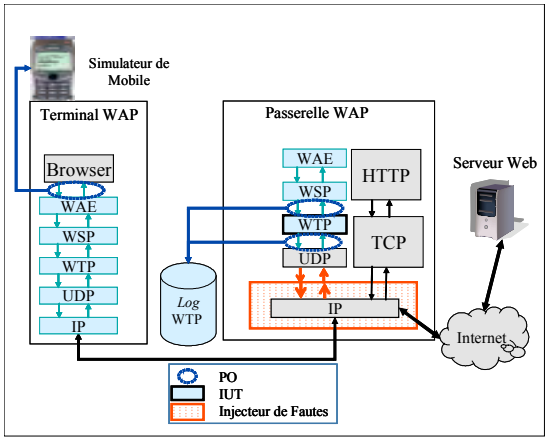


Figure 2. *L'architecture de test***.**

L'analyse des résultats est basée sur les éléments suivants: la sortie du browser vers l'utilisateur et les logs WTP collectés par les POs. Le NMB émet aussi un verdict PASS or FAIL selon que la page requise est fournie ou non en réponse. Les logs WTP sont utilisés par l'outil TestInv pour effectuer l'analyse des invariants présentés en Section 2.2.

[2] http://www.forum.nokia.com/info/sw.nokia.com

Pour les tests, étant donné notre intérêt pour les mécanismes de détection et de traitement d'erreurs spécifiés, nous nous sommes focalisé sur les services de transaction de classe 2 du WTP qui offrent des services fiables. Le scénario normal typique d'échange des messages entre le client et le serveur est montré dans la Figure 3(a), tandis qu'en 3(b) on présente le scénario en présence des fautes. Nous pouvons d'ores déjà établir un invariant simple pour vérifier ce scénario :

$$S_1 ::= RcvInvoke / \mathrm{TR}-Invoke.ind, *, \mathrm{TR}-Result.req / \{Result\}$$

Cet invariant exprime la propriété suivante : à chaque fois qu'une séquence d'entrées/sorties initiée par la paire RcvInvoke/TR-Invoke.ind est dans la trace (log WTP), alors la première occurrence de l'entrée TR-Result.req doit être suivie de la sortie Result. Le symbole "*" indique que plusieurs interactions peuvent avoir lieu entre la première et la dernière paire.

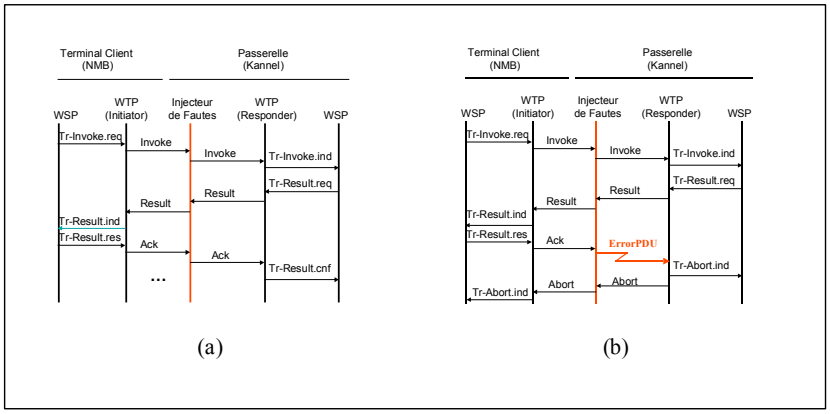


Figure 3. *Scénarios requête-réponse pour les services WTP classe 2 : (a) scénario normal ; (b) scénario avec corruption de la PDU Ack.*

Une fois que la PDU Result est envoyée, le *Responder* fait une attente temporisé pour la confirmation de la part de l'*Initiator*, ce qu'il fait en envoyant la PDU Ack. À la réception de l'événement correspondant (RcvAck), le *Responder* envoie la confirmation à la couche WSP. La propriété suivante, exprimée par un invariant d'obligation, établit que la confirmation à l'utilisateur local (TR-Result.cnf) n'est envoyée qu'après l'envoi et la confirmation de la réponse :

$$O_1 ::= ?/ Result, *, RcvAck / \overline{\{\mathrm{TR}-Result.cnf\}}$$

À présent, considérons le cas où les messages peuvent être corrompus par le canal de communication. Lorsque la couche inférieure WDP détecte un PDU erroné (erreur de *cheksum*, taille trop courte ou trop longue, etc.), un événement d'erreur RcvErrorPDU est reçu par WTP. Dans ce cas, le *Responder* doit envoyer une PDU

Abort à l'*Initiator*. Ce comportement peut être exprimé par l'invariant simple suivant :

$$S_2 ::= RcvErrorPDU / \{Abort\}$$

Un autre mécanisme de traitement d'erreur utilise le *timer* de retransmission (R). Lorsque le *Responder* transmet une réponse (PDU Result), il déclenche le timer R pour attendre la confirmation pendant un certain temps. À l'échéance de R (indiqué par l'événement TimerTO_R), le *Responder* retransmet la réponse et reprend l'attente de confirmation. Au bout d'un nombre maximum de retransmissions, la transaction est avortée et une indication d'échec est envoyée au WSP. Cette propriété est exprimée par l'invariant simple suivant :

$$S_3 ::= TimerTO_R / \{Result, \mathrm{TR}-Abort.ind\}$$

Les propriétés présentées jusqu'ici représentent uniquement les aspects fonctionnels. Or en présence de fautes, le système peut se retrouver dans des situations de blocages (*deadlocks*) où l'IUT n'accepte aucune entrée et ne produit aucune sortie, ou encore des situations où l'IUT avorte son exécution suite à une erreur non traitée. Pour cela, on a défini deux entrées supplémentaires représentant ces situations de blocage et d'avortement de l'exécution: PANIC et HANG. La sortie est considérée comme NULL dans les deux cas. Deux invariants simples (S_4 et S_5) ont été créés pour vérifier la présence de ces événements sur la trace. Notons que contrairement aux invariants présentés préalablement, la validité de S_4 et S_5 ne garantit pas la robustesse de l'IUT.

$$S_4 ::= HANG / \{NULL\}$$

$$S_5 ::= PANIC / \{NULL\}$$

Dans cet article, nous nous limiterons à la présentation de ces propriétés (S_4 et S_5) car elles permettent déjà de montrer l'utilité de l'approche. Bien sur, la notation permet aussi de représenter d'autres aspects, comme par exemple, les paramètres de primitives et PDUs. Plus de détails sont disponibles dans (Bayse 2006).

Les résultats obtenus jusqu'à présent nous permettent déjà de faire plusieurs constatations. Tout d'abord, ces résultats ont permis de démontrer que considérer uniquement les observations au niveau de l'application n'est pas suffisant pour déterminer la robustesse d'un système. En effet lors d'une expérimentation, l'application s'est terminée avec succès, sans pour cela que les propriétés soient satisfaites. Plus précisément, S_3 a eu un verdict FAIL car l'IUT envoie un message de plus que ceux prévus dans la spécification. Une autre expérience nous a permis d'observer un comportement d'arrêt sous défaillance (*fail stop*) du protocole (S_5 est vrai), ce qui est parfaitement acceptable. Enfin, le seul cas de défaillance observé et considéré comme le plus critique concerne le blocage de l'IUT (S_4 est vrai) qui entraîne le blocage du client à son tour. Cependant, les propriétés S_1 à S_3 sont satisfaites dans ce cas.

6. Conclusions et Perspectives

Dans cet article nous avons présenté une approche pour le test de robustesse combinant l'injection des fautes par logiciel avec le monitorage. À notre connaissance, aucune des approches existantes de test de robustesse ne propose une telle combinaison. L'injection des fautes permet, entre autres, un meilleur contrôle des fautes comparée aux testeurs usuellement utilisés pour les tests actif. Par opposition aux techniques d'analyse de résultats utilisées traditionnellement dans les études basées sur l'injection des fautes, le monitorage permet une analyse plus complète et plus fine des résultats.

L'approche que nous proposons utilise des invariants pour représenter les propriétés attendues de l'implantation en présence de fautes. L'utilisation des invariants permet d'analyser le comportement d'un système modélisé par des machines à états finis. Comme le modèle des machines à états finis est largement utilisé pour spécifier des protocoles de communication, l'approche est donc très utile et accessible aux utilisateurs. À travers cette étude et moyennant quelques adaptations des outils existants, nous avons montré qu'il était possible d'appliquer l'approche proposée sans grande difficulté. Dans nos travaux futurs, nous envisageons d'étendre l'approche proposée pour la prise en compte des aspects temporels qui sont difficiles à modéliser par des expressions régulières.

Remerciements

Les auteurs tiennent à remercier la CAPES pour le soutient de l'un des auteurs. Nous remercions également Amel Mammar pour ses remarques et correction du français.

7. Bibliographie

Bayse E., Cavalli A., Nunez M., Zaidi F., « A Passive Testing Approach based on Invariants: Application to the WAP », *Computer Networks*, 48, pp247-266, 2005.

Bayse E., Méthodologie de test passif par invariants. Application au protocole WAP, Thèse de doctorat, Institut National des Télécommunications en co-accréditation avec l'Université d'Evry-Val d'Essone, 2006.

Bochmann G.v., Petrenko A., « Protocol Testing: Review of Methods and Relevance for Software Testing », *Actes des* International Symposium on Software Testing and Analysis (ISSTA), August 17-19, 1994, Seattle, WA, USA, pp 109-124.

Castanet R., Waeselynk H., Techniques avancées de test de systèmes complexes: test de robustesse, rapport de recherche, 2003, CNRS-AS23.

Drebes R.J., Jacques-Silva G., Fonseca da Trindade J., Weber T.S., « A Kernel-based Communication Fault Injector for Dependability Testing of Distributed Systems », *Actes*

des Parallel and Distributed Systems: Testing and Debugging (PADTAD-3), 2005, Haifa, Israel.

Hadzilacos V., Toueg S., A modular approach to fault-tolerant broadcasts and related problems. Technical Report TR 94-1425, Cornell University, Dept. of Computer Science, Cornell University, Ithaca, NY 14853, May 1994.

Khorchef F. S., Berrada I., Rollet A., Castanet R., « Cadre formel pour le test de robustesse. Application au protocole SSL », *Actes des CFIP'2006* , Tunisie, Octobre 2006, pp199-202.

Open Mobile Alliance (OMA, ex-WAP Forum). WAP Wireless Transaction Protocol Specification, WAP-224-WTP-20010710-a, July/2001.

LaasNetExp : une plateforme expérimentale pour l'émulation et les tests en réseaux

Philippe Owezarski, Yann Labit, Pascal Berthou, David Gauchard

LAAS-CNRS
Université de Toulouse
7 Avenue du Colonel Roche - 31077 Toulouse cedex 4, France
{owe, ylabit, berthou, gauchard}@laas.fr

RÉSUMÉ. *L'expérimentation en réseaux est incontournable pour évaluer et valider de nouvelles technologies, architectures et protocoles de communication. Il s'avère que les résultats obtenus à partir de simulations sont la plupart du temps irréalistes, les simulateurs ne pouvant pas intégrer tous les modèles de comportement des composants réseaux, des systèmes opératoires et des applications des machines d'extrémité. Des travaux autour de l'émulation et de l'expérimentation réelle en réseaux ont donc vu le jour. Ce papier présente les motivations et besoins qui ont conduit à la mise en place d'une telle plate-forme expérimentale au LAAS - LaasNetExp - et décrit ses principes de conception. L'article détaille comment les conditions expérimentales peuvent être contrôlées pour des expériences reproductibles et faciles à analyser. L'article montre aussi comment des conditions expérimentales réalistes (configuration des émulateurs et générateurs de trafics) peuvent être mises en œuvre à partir des résultats de caractérisation et analyse du trafic Internet. Cet article évalue le réalisme des expérimentations ainsi obtenues.*

ABSTRACT. *Network experiments are essential for assessing and validating new networking technologies, architectures and protocols. These assessments have long been performed using network simulators. But it clearly appeared that the results got in simulations are not realistic, simulators being unable to accurately integrate all models of all networking components, end host operating systems and applications. Therefore, some work has been issued for developing real experiment platform and network emulators. This paper addresses the motivations that raised the setting-up of such an experimental platform at LAAS - LaasNetExp - and describes its design. It is detailed how experimental conditions can be fully controlled for reproducible and easy to analyze experiments. This paper also describes how realistic conditions can be set-up in experiments (configuration of emulators and traffic generators) based on the results of Internet traffic characterization and analysis. The realism of such experiments is thus assessed.*

MOTS-CLÉS : *Expérimentation en environnement réel, émulation, expérimentations réalistes, expérimentations reproductibles et contrôlées, générateurs de trafic*

KEYWORDS: *Experiments in real environment, emulation, realistic experiments, reproducible and controlled experiments, traffic generator*

1. Motivations

La recherche en réseaux ne peut pas se dissocier d'expérimentations faites soit à l'aide de simulations, soit par émulation ou enfin par expérimentations en environnements réels. Ces expérimentations sont essentielles pour évaluer et valider a priori des architectures et/ou protocoles de communication en cours de conception.

En simulation, tous les composants du système distribué (i.e. les réseaux interconnectés et toutes les machines d'extrémité) sont représentés par des modèles ([FAL 99]) dont le réalisme peut toujours être discuté. De fait, dès que l'on considère le réseau Internet, les simulations deviennent difficiles [FLO 01] notamment à cause de sa taille (nombre d'utilisateurs et d'équipements), de sa complexité (le nombre de protocoles différents par exemple), des comportements très variables de ses utilisateurs, mais aussi des évolutions rapides de ses technologies et usages... On comprend aisément dans pareilles circonstances pourquoi il est si difficile de réaliser des simulations réalistes de l'Internet. Une des limites des simulateurs est leur extensibilité limitée : simuler ne serait-ce qu'une fraction de l'Internet nécessite des machines aux puissances de calcul et capacités mémoire difficilement accessibles.

Même si la simulation est toujours l'outil le plus utilisé dans la Recherche en réseau (notamment pour son faible coût en termes d'investissements matériel et humain - cf. figure 1.a), il est généralement constaté que les résultats sont très imprécis. Les différences entre les résultats de simulations et ceux obtenus en environnement réel sont généralement significatives. Ainsi, des plates-formes d'expérimentation en environnement réel ont aujourd'hui le vent en poupe (comme Planetlab - cf. les figures 1.a et b). Sur de telles plates-formes, tous les composants, les systèmes opératoires, les protocoles, etc. sont réels. Toutefois, le problème lors de l'utilisation de ces plates-formes vient de l'impossibilité de contrôler les conditions expérimentales - le trafic de fond par exemple - ce qui rend l'analyse des résultats délicate voire impossible. Avec Planetlab, par exemple, qui n'est pas complètement monitorée, il est souvent impossible d'analyser les causes (possiblement externes) des problèmes observés sur les protocoles que l'on teste. De plus, ces expérimentations ne sont pas reproductibles par la non maîtrise des conditions expérimentales.

Le compromis adapté pour réaliser des expérimentations réalistes et exploitables paraît être l'émulation de réseaux. En émulation, les machines d'extrémité, leurs OS, applications et les protocoles de bout en bout sont réels. Seul le comportement réseau est simulé. Cette solution permet des expérimentations dans lesquelles la totalité des conditions expérimentales sont contrôlées, tout en minimisant les coûts d'équipements par rapport aux plates-formes en environnement réel. De plus, un émulateur peut être configuré pour se comporter comme un routeur (émulation fine) ou comme un système autonome complet, ce qui permet d'adapter la granularité d'émulation en fonction des besoins, et donc de résoudre les problèmes d'échelle des scénarios expérimentaux.

Les figures 1.a et b illustrent les capacités des différentes familles d'outils pour l'ingénierie réseau à réaliser des expérimentations réalistes, et ce en fonction du coût induit.

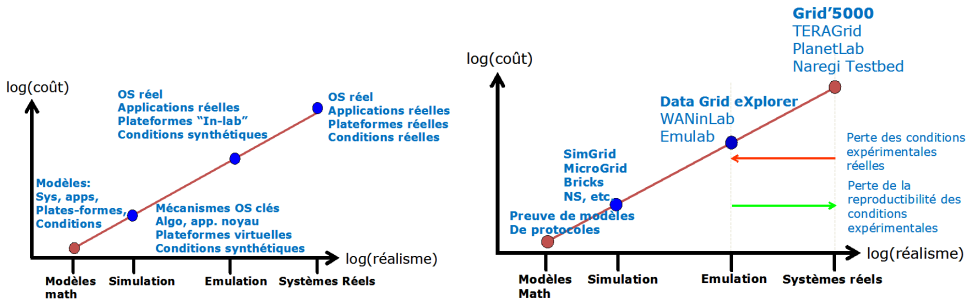


Figure 1. *Les outils d'ingénierie réseau. (a) Niveaux de virtualisation, de réalisme et de coût des différentes familles d'outils. (b) Exemples d'outils dans chacune des familles.*

Dans ce contexte, une plate-forme d'expérimentation a été conçue et installées au LAAS. L'objectif pour cette plate-forme est de répondre au plus grand nombre de demandes, et donc d'être aussi générique que possible. Elle a été conçue pour permettre à la fois de réaliser des expérimentations en émulations réseau, mais aussi en environnement réel. De plus, cette plate-forme permet une utilisation simultanée pour plusieurs expérimentations différentes et reproductibles. Cette plateforme a été baptisée "LaasNetExp" ("LAAS Network Experiments").

Le papier est organisé de la manière suivante : La partie 2 présente en détails les besoins des chercheurs avec cette plateforme expérimentale. La partie 3 décrit comment elle a été conçue et mise en œuvre, et comment elle est gérée. Enfin, dans la partie 4, il est montré comment des conditions expérimentales réalistes (configuration des émulateurs et générateurs de trafics) peuvent être mise en œuvre à partir des résultats de caractérisation et d'analyse du trafic Internet, et évalue le réalisme des expérimentations ainsi obtenues.

2. Besoins expérimentaux

Après avoir présenté les motivations de création de LaasNetExp, cette partie présente les différents atouts de cette dernière. LaasNetExp a été définie pour satisfaire aux quatre besoins principaux :

– **Un contrôle expérimental total** : Dans le but de mener des expérimentations exploitables et reproductibles, il est primordial d'avoir un contrôle total sur les conditions expérimentales, notamment en termes de trafic (charges, propriétés, profils...). Le contrôle total signifie aussi de pouvoir gérer automatiquement des (re-)configurations de la plate-forme sans l'arrêter, surtout lorsque que cette dernière est partagée par plusieurs expérimentations : les expérimentations doivent être indépendantes même quand elles partagent toutes ou parties des ressources.

– **Monitoring et mesure en continu des expérimentations** : LaasNetExp doit fournir toutes les mesures nécessaires sur les expérimentations en vue de leur analyse, évaluation et validation. Le système de monitoring et de mesure doit être transparent, i.e. ne pas influer sur les expérimentations en cours. Il doit de plus être très précis (avec des estampilles de grande précision), fiable (aucun paquet ne doit être manqué par exemple) et capable de stocker et analyser des fichiers de traces de grandes tailles sans perturber le déroulement des expérimentations en cours.

– **Intégration avec d'autres plateformes** : un des problèmes essentiels avec les réseaux aujourd'hui a trait à leur taille. Les problèmes d'échelle peuvent se régler facilement en émulation, en jouant sur la granularité des émulations (un émulateur peut émuler le comportement d'un unique composant réseau, ou d'un réseau tout entier). Par contre, en environnement réel, il est nécessaire de pouvoir disposer de plus de machines. Pour ces raisons, il est nécessaire que LaasNetExp puisse s'interconnecter avec d'autres plates-formes expérimentales, tout en gardant à l'esprit que les conditions expérimentales doivent être contrôlées et monitorées.

– **Isolation du monde extérieur ou entre expérimentations différentes** : pour pouvoir contrôler les conditions expérimentales, il faut s'isoler du monde extérieur, de son trafic et plus spécifiquement de ses anomalies. Notre plate-forme représentant une puissance de calcul et de communication importante, il faut également la protéger des intrusions. L'isoler de l'extérieur garantit un certain niveau de sécurité, mais il doit être renforcé par d'autres mécanismes de sécurité (filtrage par exemple, car nous connaissons à l'avance les caractéristiques des trafics expérimentaux venant de l'extérieur).

3. Description de la plateforme LaasNetExp

La figure 2 présente la plateforme LaasNetExp dans sa globalité. Pour remplir les besoins expérimentaux, LaasNetExp est complètement séparée du réseau opérationnel du LAAS afin d'éviter les perturbations mutuelles. LaasNetExp est reliée à l'Internet directement par RENATER, pour pouvoir profiter des nombreux services IP qui sont offerts par GEANT, RENATER et la plupart des réseaux de la recherche européens. LaasNetExp n'est donc pas connectée au réseau régional Midi-Pyrénées Rémip (ce qui aurait dû être le cas), car Rémip est un réseau commuté de niveau 2 qui n'offre donc pas les services IP dont nous avons besoin (cf. suite de cette partie).

Actuellement, LaasNetExp est composée d'un serveur et de 38 machines d'expérimentation (PC Dell PowerEdge 860, processeur Xeon Core 2, à 2.13 GHz, 2 Go de RAM et disque dur de 600 Go) fonctionnant sous différents OS et équipés de quatre interfaces Ethernet. Pour permettre des expérimentations en émulation et en environnement réel, deux réseaux ont été crées dans LaasNetExp : un réseau public réel de 3 domaines (pour les expérimentations multi-domaines) associé à des adresses IP publiques déclarées comme appartenant à 3 réseaux différents, et un réseau d'émulation. Chaque machine a deux interfaces Ethernet (Eth0 et Eth1) associées aux adresses IP publiques (Eth0 dans le domaine1 et eth1 soit dans le domaine2 soit dans le domaine3)

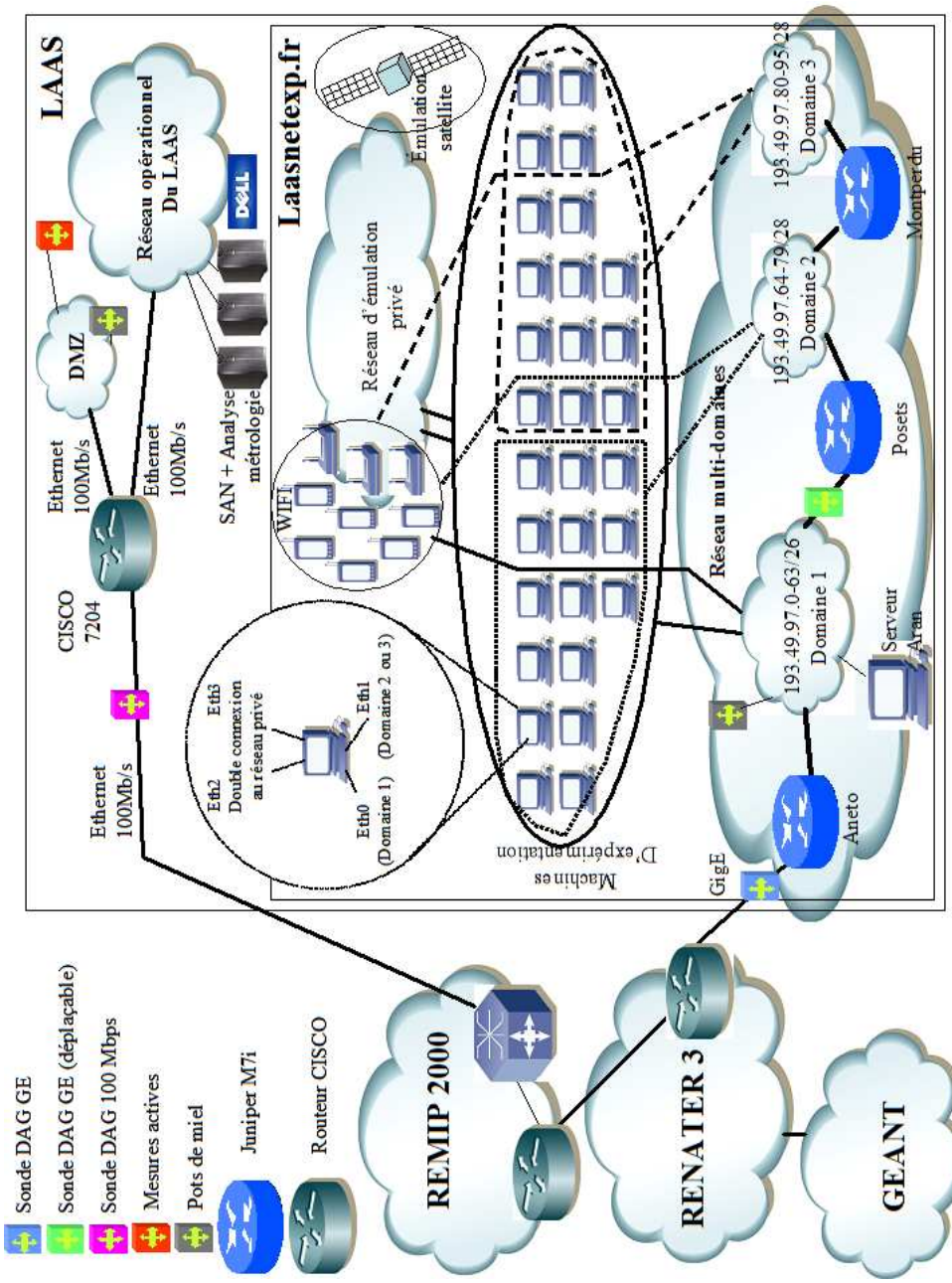


Figure 2. *La plate-forme LaasNetExp*

et les deux autres interfaces Ethernet (Eth2 et Eth3) liées à deux adresses privées dans le réseau d'émulation.

3.1. *Le réseau d'expérimentation réel*

Concernant le réseau réel multi-domaines de LaasNetExp, chaque domaine est implémenté par un commutateur Ethernet Cisco Catalyst 4948-10GE avec 24 ports Gigabits. Les trois domaines sont interconnectés par des routeurs Juniper M7i, nommés *aneto*, *posets* et *montperdu* (cf figure 2). *Aneto* est le routeur d'accès à RENATER, ce qui signifie qu'il est un point sensible et ne subira pas de conditions expérimentales risquées. La suite de cette partie détaille et justifie les choix faits pour LaasNetExp pour satisfaire aux besoins expérimentaux.

Reproductibilité des expérimentations grâce au contrôle du trafic et de la charge des machines. LaasNetExp est isolée du réseau du LAAS. Il ne passe donc sur ce réseau que le trafic généré par nos expérimentations. De plus, on peut renforcer cette propriété grâce au routeur *aneto* - en contact avec l'extérieur - en lui faisant filtrer le trafic entrant potentiel qui ne serait pas lié aux expérimentations en cours. De plus, avec un planning strict des machines, on évite d'avoir des machines utilisées simultanément dans deux expériences. Par conséquent, charge des machines et trafic sur le réseau sont totalement maîtrisés, assurant ainsi la reproductibilité des expérimentations.

Mesures et monitoring. Le système de mesure et monitoring mis en place s'appuie sur la carte DAG d'ENDACE [CLE 00]. Située derrière un splitter électrique ou optique qui laisse passer 80% de la puissance sur le lien normal, et en récupère 20% pour la carte DAG en dérivation, le trafic n'est absolument pas perturbé. Ce système de monitoring est complètement *transparent*.

La carte DAG extrait ensuite en temps réel l'entête de tous les paquets circulant sur le lien, y ajoute une estampille GPS de haute précision (64 bits) et stocke le tout dans un fichier sur le disque dur local. La machine est un serveur Dell PowerEdge 1950, avec un processeur Xeon à 1,6 GHz, 2 Go de RAM et 1To de disques durs. Elle dispose aussi d'un bus PCI étendu de 64 bits de large et de fréquence supérieure à 66 MHz. Avec une telle configuration, il est garanti qu'aucun paquet ne sera manqué. De plus, l'horloge GPS assure une synchronisation de toutes les sondes de capture avec une précision inférieure à 2 μs sur le temps de référence universel. La solution DAG est donc *fiable* et *précise*.

Enfin, un SAN (Storage Area Network) de 4,2 To et 5 serveurs de calcul (Dell PowerEdge 6850, avec 2 processeurs Xeon à 3,2 GHz, 32 Go de RAM et 1,5 To de disques durs) ont été installés sur le réseau opérationnel du LAAS et permettent d'analyser les traces collectées. Naturellement, les transferts entre machines DAG et machines d'analyse se font lorsqu'il n'y a aucune expérimentation en cours.

Un système DAG a également été installé à la sortie du réseau opérationnel du LAAS pour nous permettre de capturer des traces de trafic réel.

Intégration avec d'autres plates-formes. Par rapport aux problèmes d'échelle des expérimentations, il est possible d'interconnecter LaasNetExp avec d'autres plates-formes similaires. Toutefois, pour conserver le contrôle des conditions expérimentales, l'interconnexion se fait par des tunnels PIP (Premium IP) offerts par GEANT et la plupart des réseaux nationaux de la recherche qui garantissent que les paquets PIP seront routés en priorité. Grâce à la bande passante réservée et ce niveau de priorité maximal, il n'y a pas de perte ni de gigue observable sur ce service. PIP donne l'impression d'être seul sur un réseau réservé, et offre donc des conditions reproductibles.

Finalement, PIP cache la complexité des interconnexions de GEANT et des réseaux nationaux de la recherche qui apparaissent comme un unique domaine libre. La structure multi-domaines du réseau expérimental est donc celle de LaasNetExp qui est complètement contrôlée et permet des expérimentations reproductibles.

Isolation des différentes expérimentations entre elles. Pour isoler plusieurs expérimentations en cours simultanément, il faut s'assurer qu'aucune machine n'intervient dans les deux expérimentations : cela se fait par un système de planning de réservation. Ensuite, il ne faut pas que les trafics des 2 expérimentations interagissent. Pour cela, nous utilisons des VLAN qui permettent de créer différentes routes statiques et disjointes. Naturellement, ceci est possible et aisé grâce aux capacités largement surdimensionnées de la plate-forme.

Gestion. Toutes les fonctions de gestion du réseau public de LaasNetExp sont concentrées sur le serveur *aran* : DNS, FTP, Web, comptes des développeurs, etc.

Aran héberge également un serveur PXE. PXE utilise une base de données dans laquelle sont stockées les images des différents OS utilisés par les différentes expérimentations, et que les utilisateurs de LaasNetExp peuvent télécharger et installer automatiquement sur les machines qui leur ont été attribuées pour une expérimentation. Ainsi, chaque fois qu'une nouvelle version ou configuration d'OS est nécessaire pour une nouvelle expérimentation, elle est stockée sur le serveur et donc facilement ré-installable. Grâce à ce serveur, il est possible de facilement mutualiser la plate-forme pour de nombreuses expérimentations tout en ne perdant pas beaucoup de temps en reconfiguration des OS (et des topologies de VLAN au niveau des commutateurs Ethernet).

3.2. *Le réseau d'émulation*

Le réseau d'émulation de LaasNetExp est un réseau privé avec des adresses privées non routables. Il consiste en un routeur CISCO Catalyst 6504 avec 96 ports GigaEthernet, et une matrice de commutation non bloquante. La suite, présente les différents choix faits pour ce réseau d'émulation.

Contrôle expérimental de bout en bout. Par construction, le réseau est privé et son trafic totalement contrôlé. On se réfère également au planning de réservation des différentes machines pour éviter qu'une des machines soit impliquées dans plusieurs expériences. La charge des machines est donc contrôlée aussi. Toutes les conditions expérimentales sont donc sous contrôle, et les expérimentations, par conséquent, reproductibles.

Mesures et monitoring. La même solution DAG que celle décrite précédemment est utilisée.

Intégration avec d'autres plates-formes. Pour l'instant, cet aspect ne s'applique pas à la plate-forme d'émulation qui est entièrement privée. Toutefois, il serait facile d'utiliser une des machines comme un routeur entre les espaces privé et public, et d'utiliser des tunnels PIP pour interconnecter la plate-forme d'émulation avec d'autres plates-formes similaires.

Isolation entre expérimentations simultanées. Pour isoler des émulations différentes (qui utilisent des ensembles disjoints de machines), l'utilisation de VLAN appropriés et les capacités de communication surdimensionnées assurent que les deux expériences ne vont pas nécessiter des ressources communes, et n'ont donc aucune relation de dépendance l'une envers l'autre.

Emulation avec des technologies réseaux spécifiques. Par rapport à nos travaux sur les réseaux satellites, la plate-forme d'émulation est utilisée pour émuler des interconnexions particulières entre des réseaux satellites et terrestres. De même, la plate-forme intègre des périphériques sans fils (mini PC, PDA) et un accès WIFI.

Gestion. Comme pour la partie publique de la plate-forme d'expérimentation, il est possible de dynamiquement configurer les machines et le réseau impliqué dans une expérimentation particulière. Pour l'OS, le serveur PXE est utilisé comme déjà indiqué. Pour la configuration réseau, on utilise des VLAN. Chaque expérimentation définit son ensemble de VLAN pour émuler artificiellement la topologie sur le commutateur CISCO Catalyst 6504. Le script permettant de configurer la topologie (l'ensemble de VLAN) permet alors de facilement et rapidement restaurer l'environnement spécifique d'une expérimentation donnée. Cela permet ainsi d'utiliser simultanément la plate-forme pour plusieurs expérimentations sans perdre beaucoup de temps en reconfiguration.

4. Utilisation de LaasNetExp pour des expérimentations réalistes et reproductibles

Par construction, LaasNetExp offre un environnement dans lequel les conditions expérimentales sont complètement contrôlées et reproductibles. Il ne reste donc plus qu'à faire en sorte que ces conditions soient réalistes, ce qui est essentiel lors du test de n'importe quel protocole ou architecture de communication. De même, il est souvent intéressant de les tester dans des conditions extrêmes pour évaluer leurs limites.

A partir des traces de trafic capturées sur le réseau opérationnel du LAAS ainsi qu'à partir d'autres traces publiques, il est possible d'extraire les distributions caractéristiques pour les délais, les pertes, etc. dans l'Internet. Il est alors possible de configurer les émulateurs pour qu'ils reproduisent ces distributions.

4.1. *Génération de trafic réaliste*

Il reste cependant à générer du trafic réaliste sur le réseau expérimental, et notamment le trafic de fond. Par trafic réaliste, nous voulons dire du trafic qui possède toutes les caractéristiques statistiques du trafic réel, à savoir une variabilité forte, de la dépendance longue (LRD), des structures de corrélations spécifiques, etc. [PAR 00] [WIL 98]. Il a en effet été démontré que ces propriétés avaient un impact négatif sur la qualité de service des réseaux [PAR 97], les pics ou les variations fortes du trafic n'étant pas aisés à gérer pour les protocoles ou les mécanismes réseaux actuels. Ce sont ces "mauvaises" propriétés du trafic qu'il faudra également exagérer pour des évaluations aux limites.

Pour générer un tel trafic, nous avons défini un modèle Gamma-Farima capable de décrire les propriétés de variabilité, de corrélation et de LRD du trafic normal, mais aussi du trafic contenant tous types d'anomalies. De nombreux modèles existaient déjà, mais la plupart utilisent des hypothèses gaussiennes (irréalistes pour le trafic Internet), et les autres sont bien trop complexes pour être facilement utilisables. Notre modèle Gamma-Farima a été présenté dans [SCH 06]. Son originalité et sa force sont sa capacité à modéliser facilement le trafic Internet qui est non gaussien et à dépendance longue. Il se base sur l'utilisation combinée d'une loi Gamma pour modéliser les distributions des lois marginales du trafic et d'un processus Farima pour caractériser sa fonction de corrélation et de LRD. Au final, toute la complexité du trafic peut s'exprimer à l'aide de seulement 5 paramètres :

– α est le paramètre de forme (loi Gamma) du trafic

– β est le paramètre d'amplitude (loi Gamma) du trafic

– d exprime la LRD (Farima)

– ϕ et θ expriment la dépendance courte (Farima)

Pour les lecteurs intéressés, le modèle $Gamma(\alpha, \beta) - Farima(\phi, d, \theta)$ et sa validation sur de nombreuses traces de trafic sont présentés dans [SCH 06].

Un générateur de trafic GFtg (pour Gamma-Farima traffic generator) a été mis au point à partir de ce modèle. Il se compose de deux outils :

– Le générateur de séries Gamma-Farina, qui à partir des 5 paramètres du modèle, génère une série temporelle indiquant le nombre d'octets ou de paquets qui doivent être générés par intervalles de temps δ ;

– L'injecteur qui injecte dans le réseau le trafic comme indiqué dans la série temporelle produite par l'outil précédent. Pour respecter les temps entre deux paquets

consécutifs générés par l'émetteur et éviter les interactions avec l'environnement réseau, le trafic est généré en utilisant UDP.

4.2. *Validation du générateur de trafic Gamma-Farima*

Méthodologie

GFtg doit générer du trafic réaliste en point à point. Il est donc seulement nécessaire de valider GFtg localement, i.e. vérifier que le trafic en sortie de la carte réseau respecte les caractéristiques souhaitées.

La validation de GFtg a été réalisée sur LaasNetExp. Un PC standard a été utilisé pour générer le trafic, ainsi qu'un système de capture de trafic DAG, connecté sur le lien entre le PC et le commutateur Ethernet.

GFtg a été validé sur plusieurs traces : des traces publiques (Auckland, NLANR, etc.) et d'autres collectées dans le cadre du projet MétroSec (ces traces sont des traces de trafic normal, mais également des traces contenant des anomalies parfois légitimes - comme des foules subites - ou illégitimes comme des attaques DoS). Le tableau 1 liste les traces utilisées dans la validation de GFtg. Il indique également le type d'anomalies générées dans le cadre de MétroSec [SCH 07].

Foules subites. Dans le cadre de MétroSec, des expériences de foules subites sur un serveur Web ont été conduites. Pour les réaliser tout en ayant une activité la plus réaliste possible, la foule subite n'a pas été générée par des automates, mais avec de vrais utilisateurs volontaires. La cible était le serveur web du LAAS qui a été consulté de façon intensive pendant une certaine période de temps. Toutefois, la façon de naviguer de chaque participant relevait de sa propre initiative.

Attaques DoS. Les attaques de DDoS ont été réalisées en utilisant soit Iperf soit Trinoo (sur des machines Linux) pour générer des flux UDP avec des débits différents. Par rapport à Iperf, Trinoo utilise un démon installé sur chaque site participant à l'attaque et permet de créer des attaques plus complexes et réalistes. Les scénarios mis en œuvre comportent 4 sites d'attaque (Mont de Marsan, Lyon, Nice et Paris) qui "bombardaient" une machine du LAAS à Toulouse, où le trafic a été collecté. Le trafic lié à cette attaque a été transmis par l'intermédiaire du réseau RENATER. L'utilisation de plusieurs outils a permis d'évoluer d'attaques de DDoS simples avec Iperf (qui n'est pas un logiciel d'attaque) vers des scénarios plus complexes dans lesquels le type des paquets, le débit, la durée, l'intensité des flots d'attaque, la taille des paquets, le débit d'émission et bien d'autres paramètres ont pu être configurés selon nos souhaits.

Pour plus d'informations sur la production de ces traces, le lecteur pourra se rapporter à [BOR 08].

Pour valider GFtg, et pour chaque trace :

– les 5 paramètres Gamma-Farina ont été calculés sur la trace ;

Données	T (s)	# Pkts	IAT (ms)	Répertoire
PAUG	2620	1	2.6	ita.ee.lbl.gov/index.html
LBL-TCP-3	7200	1.7	4	ita.ee.lbl.gov/index.html
AUCK-IV	10800	9	1.2	wand.cs.waikato.ac.nz/wand/wits
CAIDA	600	65	0.01	www.caida.org/analysis/workload/oc48/
UNC	3600	4.6	0.8	www-dirt.cs.unc.edu/ts/
METROSEC-ref	5000	3.9	1.5	www.laas.fr/METROSEC/
METROSEC-DDoS	9000	6.9	1.3	www.laas.fr/METROSEC/
METROSEC-FC	1800	3.7	0.48	www.laas.fr/METROSEC/

Tableau 1. Exemples de traces utilisées pour valider GFtg. *Paramètres généraux des traces étudiées. T est la durée de la trace, en secondes. # Pkts (10^6) est le nombre de paquets dans la trace, en millions. IAT est le temps d'inter-arrivées moyen, en ms.*

– GFtg a été utilisé pour générer le trafic correspondant à ces 5 paramètres, puis ce trafic a été capturé par le système DAG, produisant ainsi la trace de trafic rejoué ;

– Les 5 paramètres Gamma-Farina ont été calculés sur la trace rejouée et comparés à ceux de la trace originale.

Résultats de validation

Les tableaux 2 et 3 montrent pour deux exemples de traces prises parmi celles décrite dans le tableau 1 (ces deux traces ont été choisies car elles ont des débits de paquets très différents) les différences entre les 5 paramètres Gamma-Farima mesurés sur les traces originales et rejouées.

Paramètres	α	β	d	ϕ	θ
trace originale	2.56	2.40	0.222	0.407	0.172
trace rejouée	2.60	2.36	0.225	0.405	0.160

Tableau 2. *Comparaison des paramètres Gamma-Farima entre une trace originale et cette même trace rejouée avec GFtg. Le débit est de 6 paquets/ms*

Paramètres	α	β	d	ϕ	θ
trace originale	26.69	1.36	0.263	0.015	0.274
trace rejouée	29.52	1.23	0.283	0.072	0.343

Tableau 3. *Comparaison des paramètres Gamma-Farima entre une trace originale et cette même trace rejouée avec GFtg. Le débit est de 26 paquets/ms*

Il apparaît dans ces deux tableaux que la précision de l'injecteur est très bonne. Ceci a été confirmé sur toutes les traces que nous avons rejouées jusqu'à maintenant avec GFtg.

5. Conclusion

Cet article a décrit la conception et la mise en œuvre d'une plate-forme expérimentale pour conduire des recherches en réseaux. Les objectifs en termes de contrôle des conditions expérimentales pour des expérimentations reproductibles et faciles à analyser ont été atteints. L'originalité de cette plate-forme réside dans sa capacité à offrir simultanément un réseau émulé et un environnement réseau réel.

Cet article propose également des recettes méthodologiques et des outils pour des scénarios expérimentaux réalistes. Cette méthodologie exploite les résultats de caractérisation et de modélisation du trafic Internet pour configurer les émulateurs, mais aussi pour générer du trafic réaliste (des générateurs de trafic ont été spécifiquement développés). La validation de cette plate-forme et des méthodologies associées ont confirmé le réalisme des conditions expérimentales, ce qui fait de LaasNetExp un outil vraiment adapté aux expérimentations réseaux.

6. Bibliographie

[BOR 08] BORGNAT P., ABRY P., DEWAELE G., SCHERRER A., LARRIEU N., OWEZARSKI P., LABIT Y., GALLON L. AUSSIBAL J., « Une caractérisation non gaussienne et à longue mémoire du trafic Internet et de ses anomalies : validation expérimentale et application à la détection d'attaque de DDoS », *Annales des télécommunications*, 2008.

[CLE 00] CLEARY J., DONNELLY S., GRAHAM I., MCGREGOR A., PEARSON M., « Design principles for accurate passive measurement », *Proceedings of PAM'2000*, Hamilton, New Zealand, avril 2000.

[FAL 99] FALL K., « Network emulation in the Vint/NS simulator », *proceedings of ISCC'99*, July 1999.

[FLO 01] FLOYD S., PAXSON V., « Difficulties in Simulating the Internet », *IEEE/ACM Transactions on Networking*, vol. 9 (4), 2001, p. 392-403.

[PAR 97] PARK K., KIM G., CROVELLA M., « On the Effect of Traffic Self-similarity on Network Performance », *SPIE International Conference on Performance and Control of Network Systems*, novembre 1997.

[PAR 00] PARK K., WILLINGER W., « Self-similar network traffic : an overview », *Self-similar network traffic and performance evaluation, Park K. and Willinger W. eds., J. Wiley and Sons*, 2000.

[SCH 06] SCHERRER A., LARRIEU N., BORGNAT P., OWEZARSKI P., ABRY P., « Non Gaussian and Long Memory Statistical Modeling of Internet Traffic », *4th International Workshop on Internet Performance, Simulation, Monitoring and Measurements (IPS-MoMe'2006)*, Salzburg, Austria, février 2006.

[SCH 07] SCHERRER A., LARRIEU N., OWEZARSKI P., BORGNAT P., ABRY P., « Non Gaussian and long memory statistical characterization for Internet traffic with anomalies », *IEEE Transaction on Dependable and Secure Computing*, vol. 4(1), January-March 2007.

[WIL 98] WILLINGER W., PAXSON V., TAQQU M., « Self-Similarity and Heavy Tails : Structural Modeling of Network traffic », *A Practical Guide To Heavy Tails : Statistical Techniques and Applications, ISBN 0-8176-3951-9*, 1998.

Réseaux maillés

Gestion avancée de la mobilité dans les réseaux maillés sans fil

M. Bezahaf[*] — L. Iannone[**] — S. Fdida[*]

Université Pierre et Marie Curie - Laboratoire LIP6/CNRS[*]
{Mehdi.Bezahaf, Serge.Fdida}@lip6.fr

Université Catholique de Louvain - IP Networking Lab (INL)[**]
Luigi.Iannone@uclouvain.be

RÉSUMÉ. *En dépit des efforts déployés dans le domaine de la mobilité dans le contexte des réseaux maillés sans fil (Wireless Mesh Networks -WMN), la gestion de mobilité souffre toujours d'un manque de recherche. Plusieurs travaux intéressants ont été menés sur ce type de réseaux, dont quelques-uns qui proposent des solutions pour gérer la mobilité des clients. Cependant, souvent ces solutions exigent aux utilisateurs de modifier ou rajouter des modules supplémentaires dans la pile protocolaire de leur équipement. Dans ce papier, nous abordons les problèmes liés à la mobilité dans les réseaux maillés sans fil et nous proposons une solution efficace à ces problèmes. Nous concevons et implémentons* Enhanced Mobility Management, *une approche améliorée de la gestion de mobilité, qui détecte efficacement les utilisateurs lors de leurs déplacements. EMM se base sur l'utilisation du cache NDP (Neighbor Discovery Protocol) des clients afin de garder une trace de leur ancienne association au cours de leurs déplacements. Cette trace est utilisée pour mettre à jour les routes sans nécessiter l'installation de logiciels supplémentaires.*

ABSTRACT. *Despite considerable efforts, mobility management in Wireless Mesh Networks (WMN) remains an open issue. Several high performance solutions can be found in the literature, however, they all have the same requirement that refrains them from being widely adopted: they need either modifications or additional modules into the protocol stack of users' equipment. In this paper, we investigate the mobility problem in WMNs and propose a new efficient solution named* Enhanced Mobility Management *(EMM), which does not rely on any modification or additional software on the client side, thus being totally transparent for end-users. EMM takes advantage of the existing Neighbor Discovery Protocol (NDP) cache to keep a track of the last client association and uses this information to trigger an update in order to re-route packets. The measurements we performed show how EMM is able to greatly improve performances.*

MOTS-CLÉS : *Réseaux maillés sans fil, gestion de mobilité, IEEE 802.11.*

KEYWORDS: *Wireless Mesh Networks, mobility management, IEEE 802.11.*

1. Introduction

La gestion de mobilité est un paramètre très important dans les réseaux actuels, puisque les utilisateurs sont de plus en plus mobiles en raison du déploiement massif des technologies sans fil. Cette nouvelle génération de clients qui cherchent à communiquer durant leurs déplacements sans aucune contrainte de connectivité, sans aucun logiciel additionnel à installer et où le changement de réseau est complètement transparent, a poussée la communauté des chercheurs à proposer une nouvelle architecture plus adéquate à leurs besoins. Les réseaux maillés sans fil (Wireless Mesh Network - WMN) [AKY 05] font partie de cette catégorie de réseaux. Les WMNs sont une classe émergente des réseaux sans fil, capable de s'organiser et de se configurer dynamiquement. Ils prennent le principe d'un réseau sans fil basé sur la transmission multi-sauts *c.-à-d.*, les communications entre deux noeuds peuvent être supportées par plusieurs noeuds intermédiaires (appelés Wireless Mesh Router - WMR) dont le rôle est de retransmettre les informations. Leur architecture à deux niveaux concentre le routage sur une partie sans fil stable (le premier niveau - backbone), composée de WMRs qui offrent une connectivité à des clients mobiles (le deuxième niveau). Dans ce contexte, le challenge est de préserver les connexions ouvertes d'un client mobile quelque soit le type de ses déplacements. La gestion de la mobilité dans les réseaux maillés sans fil est répartie en deux phases : une phase de détection et localisation et une phase de transition. La première phase consiste à détecter le plus instantanément possible les clients lors de leurs déplacements et de déterminer leur position dans le réseau à tout moment et avec une bonne précision (*c.-à-d.*, avoir une vue réaliste du réseau). La deuxième phase complète la première en mettant à jour les informations de routage afin de réduire au minimum le temps de déconnexion, éviter la perte de paquets et maintenir les connexions déjà ouvertes. Plusieurs propositions abordent la gestion de mobilité avec de bonnes performances. Néanmoins, elles peinent à être réellement et largement déployées en raison de leur nécessité de modifier la pile protocolaire ou d'installer un logiciel supplémentaire dans l'équipement du client.

Dans ce papier, nous proposons l'approche *Enhanced Mobility Management* (EMM) pour gérer efficacement la mobilité des clients, sans avoir besoin d'installer un logiciel supplémentaire ou de modifier la pile protocolaire de ces utilisateurs. L'approche EMM est basée sur l'utilisation du cache NDP (Neighbor Discovery Protocol) du client. Plus particulièrement, un WMR injecte une entrée particulière dans le cache NDP de ses clients locaux. Cette entrée est utilisée quand un client change d'association, dans le but de reconnaître sa précédente association et de mettre à jour sa nouvelle position (nouvelle association). Les mesures que nous avons effectuées sur notre réseau de test déployé au LIP6 "MeshDVNet" ([IAN 05a], [IAN 05b]), montrent que l'approche EMM est capable d'améliorer énormément les performances de gestion de mobilité en réduisant considérablement le temps de déconnexion et donc les pertes de paquets.

Le reste du papier est organisé de la sorte. Dans la section 2, nous passons en revue les principales solutions de gestion de mobilité dans les réseaux maillés, avant d'analyser dans la section 3 le comportement original et les problèmes liés à la gestion

de mobilité dans MeshDVNet. Nous présentons alors notre approche EMM dans la section 4, suivie des mesures effectuées sur le réseau test MeshDVNet et la discussion des résultats obtenus. La Section 6 récapitule nos contributions et conclut le papier.

2. État de l'art

Les réseaux maillés sans fil [AKY 05] sont souvent sollicités pour diverses raisons : réseaux communautaires, réseaux d'entreprises ou domestiques et réseaux de secteur locaux ou métropolitains. Certains industriels ont même commercialisé des WMNs ([NORT], [CISCO], [STRIX]). L'un des plus connu est le projet MIT Roofnet [BIC 05]. L'étude Roofnet est plus ciblée sur la maintenance et l'optimisation des routes que sur la mobilité des clients. Il existe aussi quelques communautés de WMN telles que NYC wireless [NYCW] et Quail Ridge Wireless Mesh Network [MOH 07].

Dans un tel contexte, le déplacement d'un client abouti souvent sur un changement de sous-réseau, par conséquent l'adresse IP du client n'aura aucune signification dans le réseau visité. Le fait de changer l'adresse IP du client à chaque changement de Point d'Accès n'est pas une bonne solution, car toutes les connexions TCP seront perdues et toutes les applications stockant l'adresse IP souffriront. Pour résoudre ce problème, différentes solutions ont été présentées, telles que Mobile IP ([PER 02], [JOH 04]), HIP [HEN 07] et le mécanisme RendezVous [EGG 04]. Ces approches, permettent aux clients de maintenir la même identité (Identifiant ou adresse IP) pendant la visite d'autres sous-réseaux, comme elles permettent aussi leur localisation. Depuis sa conception, plusieurs améliorations ont été apportées à la version originale de Mobile IP. On peut citer HMIP [CAS 00], qui consiste à hiérarchiser ses agents pour éviter un surcoût de signalisation dans les cas où un client s'éloigne de son réseau mère ; comme on peut citer FMIP [KOO 05], qui consiste à anticiper le déplacement d'un client afin d'obtenir sa nouvelle adresse temporaire avant son déplacement réel. Dans iMesh ([NAV 05], [NAV 06]), les auteurs utilisent une variante de Mobile IP (TMIP) qui est basée sur un serveur centralisé (MLR). Dans d'autres solutions comme HAWAII [RAM 02] et Cellular IP [CAM 00], le client lui-même envoie un paquet de mise à jour à la passerelle du réseau après un déplacement, ce qui permet aussi aux routeurs sur le chemin de mettre à jour leur table de routage. SMesh [AMI 06] utilise le protocole DHCP pour avoir la position des clients. Une adresse IP est attribuée à un client pour une durée de validité de 2 secondes "bail". A expiration du bail, le client transmet une requête DHCP afin de renouveler son bail. Cette requête permet aux APs de positionner le client.

Néanmoins, notre objectif principal est d'avoir un mécanisme efficace de récupération de connexion quand un client se déplace, sans la nécessité d'une pré-installation pour ce dernier, ce qui n'est pas le cas dans la plupart des solutions existantes et les protocoles qui souvent poussent les clients à faire une installation logicielle supplémentaire.

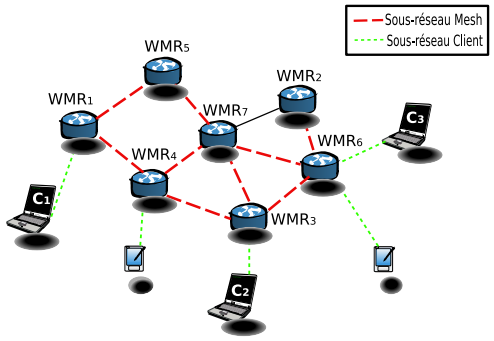


Figure 1 – Exemple de déploiement de MeshDVNet

3. Gestion de la mobilité dans MeshDVNet

Afin d'acquérir une compréhension plus profonde des problèmes de mobilité existants dans les WMNs et pour avoir une étude réaliste, nous avons analysé et effectué plusieurs mesures sur le réseau MeshDVNet ([IAN 05b], [IAN 05a]).

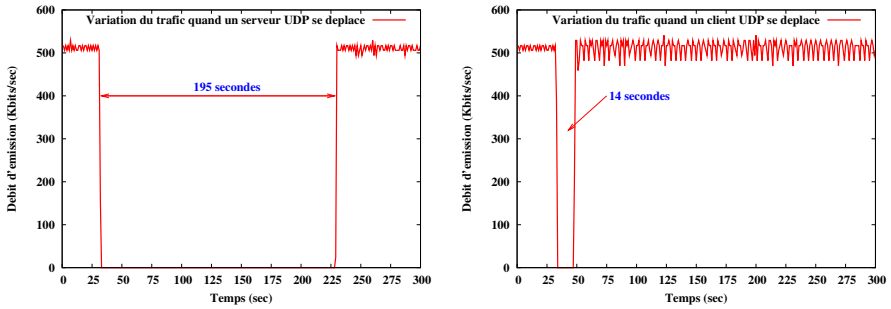
Basé sur IPv6, le réseau MeshDVNet est un WMN de test déployé au LIP6 qui offre une connexion sans fil aux clients et leur permet de communiquer sans aucune pré-installation nécessaire. Pour avoir une vue en temps réel de notre réseau de test et pour contrôler le bon fonctionnement des WMRs, nous utilisons la page Web de supervision qui est publiquement disponible (seulement pour des connexions IPv6) au :

http ://www.infradio-jussieu.lip6.fr/supervision/supervision-mesh-kennedy.html

MeshDVNet est découplé en deux sous-réseaux (Fig. 1) : un sous-réseau formé d'un ensemble de WMRs qui constitue le backbone et un deuxième formé d'un ensemble de clients. Nommés MeshDVbox, les routeurs (WMRs) utilisés dans MeshDVNet sont des Soekris net4521 sous le système d'exploitation Linux (Crux comme distribution), sur lequel le protocole MeshDV tourne. MeshDVbox est équipé de deux interfaces sans fil, une pour communiquer avec les autres WMRs et une pour communiquer avec ses clients locaux.

Afin de comprendre comment la mobilité est gérée dans la version originale de MeshDVNet et pour souligner les problèmes actuels, nous présentons ci-dessous comment une communication entre deux clients, associés à des WMRs différents, est établie. Nous supposons que le client C_1, associé à WMR_1, ouvre une connexion vers le client C_2 qui est associé à WMR_3 (Fig. 1). La communication est établie de la façon suivante :

1 : En utilisant un serveur DNS, le client C_1 peut récupérer l'adresse IP du client C_2 s'il ne la connaît pas. De cela et du fait que C_1 et C_2 sont dans le même sous-réseau logique, le client C_1 envoi d'abord un



(a) Quand le serveur se déplace (b) Quand le client se déplace

Figure 2 – Performances de mobilité dans MeshDV lors d'un trafic UDP

Neighbor Solicitation (NS) en multicast afin d'obtenir l'adresse MAC du client C_2. En recevant ce paquet de sollicitation, le WMR_1 envoi un message MCREQ[1] aux autres WMRs pour découvrir où est-ce que le client C_2 est associé.

2 : Le WMR_3 reçoit la demande en multicast (MCREQ) et répond avec un paquet CRREP[2].

3 : Quand le WMR_1 reçoit le message CRREP, il répond avec un Neighbor Advertisement (NA) au Neighbor Solicitation du client C_1 contenant l'adresse IP du client C_2 associée à l'adresse MAC du WMR_1.

4 : Par la suite, tous les paquets envoyés du client C_1 au client C_2 seront capturés par WMR_1, encapsulés dans des paquets IPv6 et envoyés à WMR_3.

5 : Le WMR_3 décapsulera les paquets reçus pour les retransmettre au client C_2.

Dans un réseau maillé sans fil, la détection des déplacements d'un client peut être réalisée de deux manières différentes : soit c'est l'ancien WMR qui détecte que son client local a changé d'association (détection automatique), soit c'est le nouveau WMR qui détecte un nouveau client et le notifie à l'ancien WMR (détection réactive). MeshDV est basé sur la première approche pour gérer la mobilité de ses clients. Si un client change d'association (*c.-à-d.* de WMR), pendant une communication déjà ouverte auparavant, il arrive à se reconnecter au niveau de la couche physique avec le nouveau WMR, mais la communication avec ses correspondants ne sera pas rétablie complètement et le problème vient de l'ancien WMR où le client était connecté.

En effet, pour ce WMR son client est toujours connecté à son interface, ce qui n'est pas le cas. Donc tous les paquets destinés à ce client seront acheminés par l'ancien WMR à une destination non existante. Ce problème est dû à la mauvaise gestion de mobilité dans MeshDV qui est basée sur la détection des clients, faite par le driver de la carte sans fil. Dans notre réseau de test, le driver utilisé par les cartes sans fil des WMRs est Madwifi [MAD], celui-ci détecte la déconnexion des clients après trois minutes de leur départ. Donc, les WMRs gardent des informations erronées de leurs

1. MCREQ (Multicast Client REQuest) est un paquet multicast, utilisé pour rechercher des clients distants, consiste à trouver le WMR qui gère le client recherché [IAN 05a].
2. CRREP (Client Request REPly) est un paquet unicast, utilisé pour répondre à une requête MCREQ [IAN 05a].

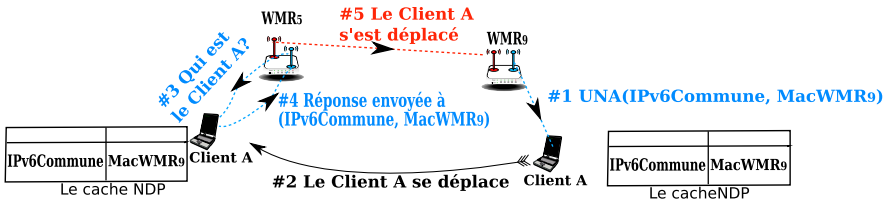


Figure 3 – Exemple de gestion de mobilité avec l'approche EMM

clients locaux qui se sont déplacés, et prennent plus de trois minutes pour se rendre compte. La réduction du timer, au niveau du driver, permet de détecter rapidement un déplacement, mais les beacons de contrôle surchargent enormement le medium.

Nous avons expérimenté différents scénarios de mobilité qui présentent quelques problèmes de performance. Nous nous sommes focalisés sur le temps de déconnexion que le client subit quand il se déplace. Pour cela nous avons utilisé trois types différents de trafic (*c-à-d.* TCP, UDP et Ping) entre deux clients pendant 300 secondes et avons observé le temps de déconnexion quand l'un d'entre eux se déplace. Notez que nous utilisons le programme Iperf [TIR 05] pour générer le trafic TCP et UDP. Comme clients nous avons utilisé des ordinateurs portables Compaq nx7000, sous le système d'exploitation Fedora Core 3. La Figure 2 représente le débit d'émission pendant chaque seconde. Quand le serveur Iperf se déplace pendant un trafic UDP (Fig. 2(a)), nous avons obtenu un temps de déconnexion de trois minutes (détection automatique). Dans le cas où le client Iperf se déplace (Fig. 2(b)), ce dernier se connecte instantanément au nouveau WMR et continu d'envoyer des paquets UDP avec l'adresse MAC destination égale à l'adresse MAC de l'ancien WMR. Cela est dû au cache du protocole NDP (Neighbor Discovery Protocol) [NAR 98], qui n'est pas rafraîchi immédiatement après la connexion physique. Ces deux scénarios sont très représentatifs, puisque le ping et le trafic TCP présentent un comportement très semblable à celui observé dans le premier cas [BEZ 07]. Dans la section suivante, nous proposons notre solution, basée sur le cache NDP, qui résout efficacement tous les problèmes vus dans cette section.

4. L'approche Enhanced Mobility Management

Enhanced Mobility Management (EMM) est une approche améliorée de la gestion de mobilité des clients, où la détection de leurs déplacements est faite entièrement par le nouveau WMR (détection réactive). Notre approche est basée sur le cache NDP qui contient essentiellement les adresses IP des correspondants avec qui un client communique, l'adresse MAC de la passerelle pour les atteindre et inclut aussi des informations sur l'état de leur joignabilité.

Le routeur modulaire Click [KOH 00] est utilisé comme infrastructure logicielle de routage. Dans notre implémentation, chaque WMR utilise le module Click au ni-

veau noyau. Cela signifie que le traitement et le filtrage de tous les paquets reçus sont effectués au niveau Click au lieu qu'ils soient fait au niveau application. Dans l'approche EMM, chaque WMR du réseau, au démarrage, rajoute la même adresse IPv6 en lien locale nommée *Adresse Commune* à son interface client. Cette adresse est statique et peut être manuellement modifiée. Elle n'est jamais utilisée pour échanger des données mais uniquement dans le but de gérer la mobilité des clients. Le message Unsolicited Neighbor Advertisement (UNA) est utilisé pour mettre à jour le cache NDP des clients pendant leurs déplacements (modifier l'entrée de l'*Adresse Commune*).

Présenté dans le RFC 2461 [NAR 98], le message UNA est utilisé par un noeud pour informer ses voisins directs du changement de son adresse de couche liaison. Le principe utilisé dans EMM est similaire au mécanisme de cookies dans les navigateurs Web *c.-à-d.*, c'est les clients eux mêmes qui gardent l'information de leur dernière association WMR. Pour mieux comprendre comment l'injection de ces cookies dans le cache NDP peut aider à gérer la mobilité des clients, prenons le scénario de la Figure 3. Lors de l'arrivée du client A dans le réseau, nous supposons que dans un premier temps il s'associe au WMR_9 puis, il change d'association et passe au WMR_5. Dans ce contexte, le mécanisme EMM NDP cookie fonctionne comme suit :

1 : A l'arrivée du Client A dans le réseau et après sa connexion au WMR_9, il reçoit de ce dernier un message UNA *"associe l'Adresse Commune à l'adresse MAC du WMR_9 et enregistre cette information dans ton cache NDP"*. Cela implique une mise à jour du cache NDP du Client A.

2 : Le Client A garde l'Adresse Commune associée à l'adresse MAC du WMR_9 dans son cache NDP pendant son déplacement.

3 : Le Client A change d'association et passe du WMR_9 au WMR_5. Après s'être connecté au WMR_5, ce dernier récupère l'adresse MAC du Client A et dérive son adresse IP[3]. Par la suite, il envoi un Neighbor Solicitation (NS) au Client A avec l'adresse IPv6 source égale à l'Adresse Commune et l'adresse MAC source égale à l'adresse MAC du WMR_5 *"Qui est le Client A ?"*.

4 : Le paquet NS envoyé par le WMR_5 n'est pas utilisé dans le but d'acquérir l'adresse MAC du Client A, qui est déjà obtenue de la couche 2, mais pour connaître l'ancienne association du Client A. Donc à la réception de cette sollicitation, le Client A consulte son cache NDP, où il trouve la cookie (Adresse Commune associée à l'adresse MAC du WMR9). Ainsi, il répond au WMR_5 avec l'adresse MAC destination égale à celle du WMR_9.

5 : Etant donné que l'interface sans fil du WMR_5 est configurée en mode promiscuous, le WMR_5 arrive à voir la réponse du Client A qui est récupérée par le module Click et renvoyée au niveau applicatif. A cette étape, le WMR_5 peut extraire l'adresse MAC du WMR_9 et dériver l'adresse IP du WMR_9 (en se basant sur le EUI-64). Une fois l'adresse IP de l'ancienne association du Client A connue, le WMR_5 n'a plus qu'à envoyer un message CWIT[4] au WMR_9 dans le but de notifier le déplacement du Client A. Au même moment, le WMR_5 envoi un message UNA au Client A *"associe dans ton cache NDP l'Adresse Commune à l'adresse MAC de WMR_5"*, afin de mettre à jour la cookie dans le cache NDP du Client A.

6 : Après la réception du message CWIT envoyé par le WMR_5, si le WMR_9 reçoit un paquet destiné au Client A, il le droppe et renvoi un message Client ERRor (CERR) au WMR qui a généré le paquet (WMR distant) *"le Client A n'est plus associé à moi"*.

7 : En recevant le message CERR, automatiquement le WMR distant envoie à nouveau un paquet MCREQ afin de localiser la nouvelle association du Client A.

La Figure 4 montre le diagramme temporel du mécanisme de gestion de mobilité des clients. Comme nous pouvons le voir, la Figure présente les différents paquets

3. Notez que nous utilisons le 64-bit Extended Universal Identifier (EUI-64) pour obtenir l'adresse IP à partir de l'adresse MAC ([IEE 97], [NAR 99]).
4. CWIT (Client WIThdraw) est un message envoyé en unicast, utilisé par le nouveau WMR afin d'informer l'ancien WMR du déplacement de son client [BEZ 07].

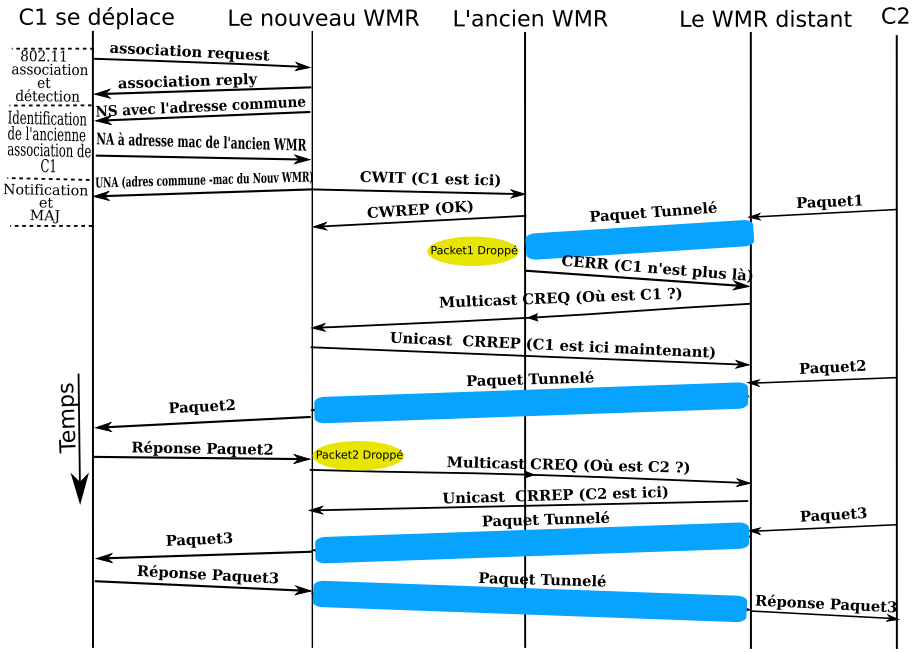


Figure 4 – Diagramme temporel de gestion de mobilité d'un client dans EMM

échangés quand le client C_1 se déplace de "l'ancien WMR" au "Nouveau WMR" durant sa communication avec le client C_2, connecté au WMR distant.

5. Evaluation de l'approche EMM

Afin d'évaluer notre travail, nous avons effectué le même ensemble de test décrit dans la section 3. Nous avons mesuré les délais de déconnexion et évalué le comportement de EMM quand les clients changent d'association. Pour un trafic UDP entre les clients, nous avons obtenu des résultats satisfaisants avec un temps de déconnexion qui est moins de la seconde (Figure 5). Dans le cas où le client qui génère le trafic UDP se déplace, le nouveau WMR doit tout à bord détecter ce client puis, faire une recherche des correspondants de ce dernier. La Figure 5 montre bien que le débit de transmission entre la seconde 233 et la seconde 234 diminue de 500 Kbits/s à 120 Kbits/s, ce qui signifie que la communication est perturbée pendant moins d'une seconde car vers la fin de la seconde la communication reprend (120 Kbits/s). De plus, entre la seconde 234 et la seconde 235 le débit de transmission augmente de 120 Kbits/s à 420 Kbits/s, ce qui signifie qu'il n'y a pas eu de perte de paquets. Notez que pour cette expérimentation, nous ne pouvons pas avoir le temps exact de déconnexion car l'outil utilisé (Iperf) ne nous permet pas d'avoir une granularité inférieur à la seconde.

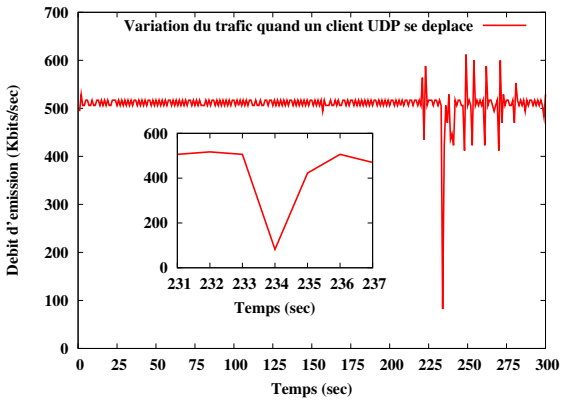
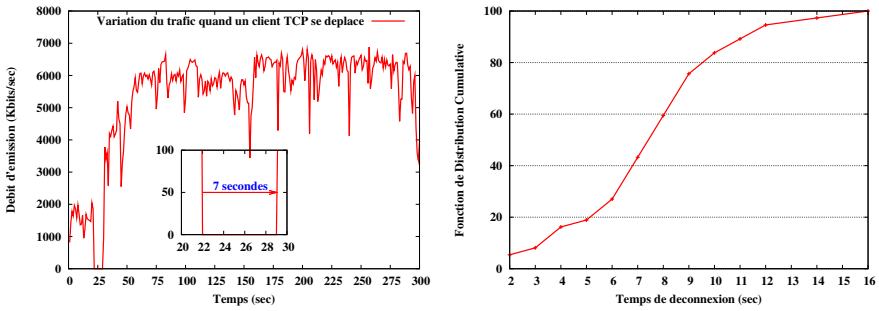


Figure 5 – Performance de l'approche EMM lors d'un trafic UDP



(a) Quand le client se déplace (b) CDF de temps de déconnexion d'un client

Figure 6 – Performances de l'approche EMM lors d'un trafic TCP

Dans le cas d'un trafic TCP entre les clients, la Figure 6(a) montre bien que les résultats ne sont plus les mêmes comparés à ceux obtenus avec un trafic UDP. Comparé au temps de déconnexion d'un client dans la version originale de MeshDV, qui était de 3 minutes, le résultat est nettement meilleur. Malgré cette amélioration, le temps de déconnexion est maintenant de quelques secondes. Cela est dû à la façon dont TCP gère la retransmission des paquets. En effet, en cas de perte de paquets, le protocole TCP utilise le timer RTO (Retransmission TimeOut) dans le but d'assurer la délivrance de ces derniers. Durant RTO secondes d'attente, si un paquet n'est pas acquitté, le RTO double et le paquet est retransmis. Quand un client se déplace et se réassocie à un nouveau WMR, ce dernier en recevant le premier paquet TCP du client ne sait pas à quel WMR est associé le correspondant de ce client. Donc, il envoie un paquet MCREQ sur le backbone et droppe ce paquet. A la fin du RTO, le paquet n'a pas été acquitté, ce qui fait que la valeur du RTO double et le paquet est retransmis. Dans ce cas là, le correspondant reçoit le paquet et l'acquitte. Cependant le WMR à

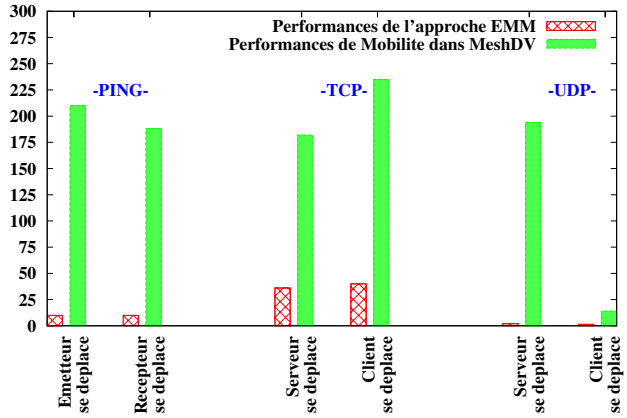


Figure 7 – Comparaison de temps de déconnexion entre la gestion de mobilité dans MeshDV et notre approche

qui le correspondant est associé (WMR distant) fait suivre l'acquittement à l'ancien WMR et non au nouveau. L'ancien WMR droppe ce paquet et notifie au WMR distant que le client à qui le paquet était destiné n'est plus associé à son interface en lui envoyant un message CERR. Cela signifie que la valeur du RTO double une deuxième fois. Ce mécanisme de retransmission, qui double à chaque perte de paquet, rajoute beaucoup de délais. De plus les cartes sans fil dans certains cas, lors du déplacement d'un client, effectuent un scan complet des canaux ce qui augmente la latence de transition IEEE 802.11 d'approximativement 6.9 secondes durant nos tests. Dans ce cas, le temps de déconnexion peut augmenté jusqu'à 30 secondes même si la détection du nouveau client et la mise à jour des tables du nouveau WMR s'effectuent instantanément après la connexion physique [BEZ 07].

La Figure 6(b) représente la fonction de distribution cumulative du temps de déconnexion d'un client en déplacement, lors d'une communication TCP et avec une transition instantanée de la couche IEEE 802.11 (*c.-à-d.* sans que le driver de la carte sans fil ne fait un scan complet des canaux). Du graphe, nous pouvons remarquer que le temps maximum de déconnexion est de 16 secondes, et que dans 95% des cas le temps de déconnexion est inférieur à 12 secondes. Nous remarquons aussi que dans quelques cas le temps de déconnexion peut être court (2 à 5 secondes).

L'histogramme de la Figure 7, représente une comparaison visuelle entre l'approche EMM et l'ancienne version de la gestion de mobilité des clients dans MeshDV. L'approche EMM diminue nettement le temps de déconnexion de 3 minutes à quelques secondes voir même moins de la seconde dans certains cas, ce qui n'est pas une amélioration négligeable.

6. Conclusions

Les solutions existantes pour la gestion de mobilité dans les réseaux maillés sans fil, constituent une sorte de paradoxe, car d'une part, elles offrent des bonnes performances, et d'autre part, elles ont des difficultés à être largement déployées dans des plateformes réelles. La raison d'un tel déficit pourra s'expliquer par le fait qu'elles exigent des installations spécifiques sur les dispositifs d'utilisateurs finaux. Le travail présenté dans ce papier représente nos efforts pour résoudre le problème de gestion de mobilité sans aucun impact, quelque soit l'équipement utilisé par l'utilisateur final. Notre proposition nommée *Enhanced Mobility Management* (EMM) est le résultat d'une analyse profonde que nous avons menée sur le réseau MeshDVNet, déployé au Laboratoire d'Informatique de Paris 6. L'approche EMM utilise les avantages du cache NDP, présent dans toutes les piles protocolaires standards, en injectant une entrée particulière (Adresse Commune). Une telle entrée est utilisée pour récupérer l'identité de l'ancienne association du nouveau client, permettant alors le re-routage des paquets et ainsi la maintenance de la communication en cours avec un temps de déconnexion relativement court. Comme nos mesures le montrent clairement, EMM comparé à la proposition originale de MeshDVNet améliore énormément les performances de gestion de mobilité pour tous types de trafic. Par exemple, dans le cas d'un trafic TCP, le temps de déconnexion est réduit à moins de 20% de la valeur originale. Même plus, dans le cas où une machine qui génère un trafic UDP se déplace, le temps de déconnexion est alors réduit à moins de 0.5% de la valeur originale. Quelques améliorations sont en cours pour éviter la perte des paquets perdus lors d'un déplacement.

7. Remerciements

Nous tenons à remercier Pierre-Emmanuel Le Roux d'avoir mis à contribution ses talents de programmeur. Ce travail de recherche à été subventionné par le Projet IST-WIP (contrat 27402) et par le Projet GigaCom.

8. Bibliographie

[AKY 05] AKYILDIZ I., WANG X., WANG W., « Wireless mesh networks : a survey », *Computer Networks - Elsevier Science*, , n° 47, 2005.

[AMI 06] AMIR Y., DANILOV C., HILSDALE M., MUSĂLOIU-ELEFTERI R., RIVERA N., « Fast handoff for seamless wireless mesh networks », *ACM Press*, , 2006, p. 83–95.

[BEZ 07] BEZAHAF M., « Fast mobility in wireless mesh networks », Master's thesis, University Pierre et Marie Curie (Paris 6), 2007.

[BIC 05] BICKET J., BISWAS S., AGUAYO D., MORRIS R., « Architecture and evaluation of an unplanned 802.11b mesh network », *International Conference on Mobile Computing and Networking (MobiCom)*, , 2005.

[CAM 00] CAMPBELL A., GOMEZ J., KIM S., VALKO A., WAN C., TURANYI Z., « Design, Implementation, and Evaluation of Cellular IP », *IEEE Personal Communications*, , 2000.

[CAS 00] CASTELLUCCIA C., « HMIPv6 : A Hierarchical Mobile IPv6 Proposal », *ACM Mobile Computing and Communication Review (MC2R)*, , 2000.

[EGG 04] EGGERT L., LIEBSCH M., « Host Identity Protocol (HIP) Rendezvous Mechanisms, draft-eggert-hip-rendezvous », IETF, Jul. 2004.

[HEN 07] HENDERSON T., « End-host mobility and multihoming with the host identity protocol, draft-ietf-hip-mm-05 », IETF, Mar. 2007.

[IAN 05a] IANNONE L., « MeshDV : Implementation Draft », *Technical Report*, , 2005.

[IAN 05b] IANNONE L., S.FDIDA, « MeshDV : A Distance Vector mobility-tolerant routing protocol for Wireless Mesh Networks », *IEEE ICPS Workshop on Multi-hop Ad hoc Networks : from theory to reality (RealMAN'06)*, , 2005.

[IEE 97] IEEE, « Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority », IEEE Standards tutorials., 1997.

[JOH 04] JOHNSON D., PERKINS C., ARKKO J., « Mobility support in ipv6, RFC 3775 », IETF, 2004.

[KOH 00] KOHLER E., MORRIS R., CHEN B., JANNOTTI J., KAASHOEK F., « The click modular router », , 2000, ACM Transaction on Computer Systems (TOCS).

[KOO 05] KOODLI R., « Fast Handovers for Mobile IPv6, rfc 4068 », IETF, juillet 2005.

[MOH 07] MOHAPATRA P., WU D., GUPTA D., « Quail Ridge Wireless Mesh Network : Experiences, Challenges and Findings », *International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, , 2007.

[NAR 98] NARTEN T., NORDMARK E., SIMPSON W., « Neighbor Discovery for IP Version 6 (IPv6), RFC 2461 », IETF, Dec. 1998.

[NAR 99] NARTEN T., « Neighbor Discovery and Stateless Autoconfiguration in IPv6 », *IEEE Internet Computing*, vol. 3, n° 4, 1999, p. 54–62, IEEE Educational Activities Department.

[NAV 05] NAVDA V., KASHYAP A., DAS S., « Design and Evaluation of iMesh : an Infrastructure-mode Wireless Mesh Network », *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, , 2005.

[NAV 06] NAVDA V., GANGULY S., KIM K., KASHYAP A., NICULESCU D., IZMAILOV R., HONG S., DAS S., « Performance Optimizations for Deploying VoIP Services in Mesh Networks », *IEEE Journal on Selected Areas in Communication (JSAC)*, , 2006.

[PER 02] PERKINS C., « IP Mobility Support for IPv4, RFC 3344 », IETF, Aug. 2002.

[RAM 02] RAMJEE R., VARADHAN K., SALGARELLI L., THUEL S., WANG S., LA PORTA T., « HAWAII : A Domain-based Approach for Supporting Mobility in Wide-Area Wireless Networks », *IEEE/ACM Transactions on Networking*, , 2002.

[CISCO] « Cisco Systems », *http ://cisco.com*.

[MAD] « Madwifi home page », *http ://madwifi.org*.

[NORT] « Nortel », *http ://nortel.com*.

[NYCW] « NYC wireless », *http ://nycwireless.net*.

[STRIX] « StrixSystems », *http ://strixsystem.com*.

[TIR 05] TIRUMALA A., QIN F., DUGAN J., FERGUSON J., GIBBS K., « Iperf-The TCP/UDP bandwidth measurement tool », 2005.

Génération de Colonnes pour le Routage et l'Ordonnancement dans les Réseaux Radio Maillés[1]

Christelle Molle — Fabrice Peix — Hervé Rivano

I3S(CNRS-UNSA)/INRIA *projet* MASCOTTE
2004 route des Lucioles, B.P. 03, 06902 Sophia Antipolis Cedex, FRANCE.
E-mail:{christelle.molle, fabrice.peix, herve.rivano}@sophia.inria.fr.

RÉSUMÉ. *Dans cet article, nous étudions la capacité des réseaux radio maillés dédiés à l'accès à Internet. Nous nous plaçons dans l'hypothèse d'un réseau synchrone fonctionnant en régime permanent où les transmissions partagent un même canal radio. Le routage et l'ordonnancement des transmissions ont un impact majeur sur la capacité fournie aux clients. Une optimisation jointe de ces deux problèmes permet de prendre en compte dans le routage les contentions dues aux interférences radio. Nous en développons des formulations exactes en programmation linéaire. Nous présentons ensuite un processus de génération de colonnes résolvant une relaxation du problème. Ainsi, nous contournons l'écueil de la combinatoire des transmissions réalisables simultanément pour permettre de calculer des solutions sur des réseaux de grande taille. Des simulations sont effectuées sur des topologies aléatoires. L'évolution de la capacité d'un réseau maillé avec ses paramètres, ainsi que la complexité algorithmique du problème sont discutées.*

ABSTRACT. *In this paper, we address the capacity of Wireless Mesh Networks dedicated to Internet access. We assume a single-channel, synchronous network in steady state. Routing and transmission scheduling have a major impact on the capacity provided to the clients. A cross-layer optimization of these problemes allows the routing to take into account the contentions due to radio interference. We develop exact linear programs for this. We then provide a column generation process computing a relaxation of the problem. It allows to work around the combinatoric of simultaneously achievable transmissions, hence computing solutions on large networks. Simulations are done on random topologies. The evolution of the capacity of a mesh network with its parameters, as well as the algorithmic complexity are then discussed.*

MOTS-CLÉS : *Réseaux Radio Maillés, Capacité, Routage, Ordonnancement, Programmation Linéaire, Génération de Colonnes.*

KEYWORDS: *Wireless Mesh Networks, Capacity, Routing, Scheduling, Linear Programming, Column Generation.*

1. C. Molle est doctorante, financée par la DGA. Ce travail est partiellement supporté par le projet européen IST/FET AEOLUS, l'ANR-JC OSERA, et l'ARC CARMA.

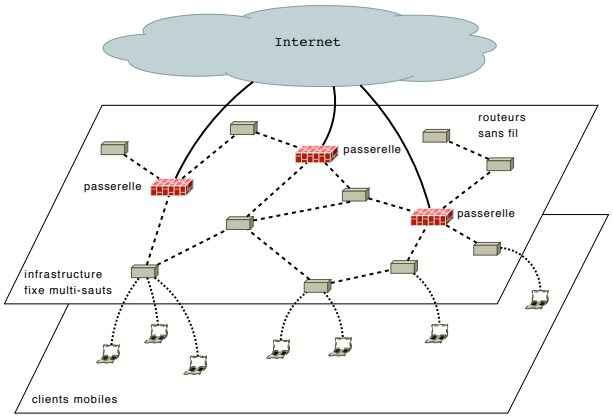


Figure 1. *Topologie d'un réseau radio maillé : les clients mobiles accèdent à Internet à travers un réseau multi-sauts composé de routeurs et de points d'accès.*

1. Introduction

Les réseaux radio maillés (Wireless Mesh Networks) sont devenus une solution efficace pour déployer à moindre coût un réseau d'accès sans fil à grande échelle, par exemple dans une ville [AKY 05]. Topologiquement, les réseaux radio maillés sont formés d'une infrastructure fixe de routeurs sans fil, reliés entre eux par des liens radio, et qui acheminent les demandes de clients mobiles vers un réseau supérieur (filaire ou non) donnant accès à Internet. Des routeurs spéciaux, appelés points d'accès, réalisent l'interconnexion entre le réseau maillé et le réseau supérieur (Fig. 1).

De nombreux travaux cherchent à concevoir des modèles d'optimization généraux pour les réseaux radio maillés afin d'évaluer les performances de ces réseaux. La capacité d'un réseau radio, définie comme étant la quantité maximale de flot que peut écouler une topologie, est au cœur de l'étude des performances des réseaux radio. En effet, le problème majeur des réseaux radio est la présence d'interférences spatiales entre les communications. Une conséquence directe de ces interférences est la réduction de la capacité [JAI 03, KOD 05]. Dans une étude analytique [GUP 00], Gupta et Kumar ont montré que la capacité d'un nœud dans un réseau ad-hoc sans fil aléatoire décroit en $\mathcal{O}(\frac{1}{\sqrt{n}})$ lorsque la taille du réseau, n, augmente. Ce résultat, prouvé dans des hypothèses simples, a été confirmé par de nombreux travaux [DOU 04, RIV 06, MEH 06]. A la différence des réseaux ad-hoc, les réseaux maillés sont fixes et possèdent des points d'étranglement, localisés autour des points d'accès, où se concentre le trafic. Par conséquent la capacité disponible en chaque nœud est réduite à $\mathcal{O}(\frac{1}{n})$ [JUN 03].

L'objectif de cet article est de pouvoir calculer une capacité optimale théorique d'un réseau radio maillé en modélisant les caractéristiques inter-couches du réseau. L'évaluation de la capacité est utile afin d'améliorer les performances des protocoles

de routage, sans oublier que l'efficacité du routage dépend lui-même d'une allocation performante des ressources physiques et techniques. En effet, les interférences radio présentes entre les transmissions doivent conditionner le routage. De même, le routage choisi doit prendre en compte la réussite des transmissions.

Afin de maximiser la capacité d'un réseau radio maillé dont nous connaissons la topologie, nous allons chercher à allouer optimalement les ressources communes, notamment l'accès au canal radio. Cette allocation est contrainte par les interférences produites autour d'un nœud émetteur : deux transmissions simultanées doivent être éloignées de plusieurs sauts. Les requêtes en contention se partagent l'accès à la ressource radio selon un multiplexage en temps. L'allocation optimale des ressources est donc un ordonnancement de l'activation des liens rendant disponible suffisament de capacité pour acheminer un trafic maximal des nœuds vers Internet [LIU 05].

L'hypothèse la plus forte de nos travaux est de considérer un réseau synchrone et une période de temps découpée en intervalles de durée fixe. Chaque lien du réseau est activé durant un certain nombre d'intervalles, ce qui entraine la création d'une planification indiquant à quel moment un lien transmet. Au cours de chaque intervalle, un ensemble de liens n'interférant pas deux à deux est activé. Cet ensemble est appelé *stable*, et l'ensemble de tous les stables possibles est exponentiel. Un ordonnancement optimal ne peut donc être calculé par des méthodes simples que sur des topologies de taille réduite. Par conséquent des études ont cherché à réduire l'ensemble des stables à considérer [WAN 07], ou développer des algorithmes d'approximations [KLA 04].

La génération de colonnes est une technique sophistiquée de résolution de problèmes linaires de grande taille [LUB 05]. Lorsque le nombre de variables est exponentiel, elle permet de décomposer le problème en différentes parties. Un programme dit *maître* résout le problème sur un sous-ensemble de variables associées à des structures combinatoires. Ces structures sont générées par des sous-programmes dits *auxiliaires*, qui sélectionnent celles qui amélioreront la solution du maître. Cette technique a déjà été utilisée [ZHA 05, CAR 07], mais pour résoudre un problème fondé sur des contraintes de routage simplifiées, se coupant la possibilité de trafic multi-chemins [LAK 06, TAM 07] exploitant plusieurs passerelles par source.

Ce travail expose deux formulations exactes du problème de routage et d'ordonnancement dans les réseaux radio maillés (section 3). Un processus de génération de colonnes est ensuite présenté (section 4). Il intègre à la fois la génération de stables et de chemins. Ce processus est enfin validé sur un ensemble d'instances aléatoires (section 5). L'évolution de la capacité de ces réseaux et la complexité algorithmique du problème sont discutées.

Dans la section suivante, nous détaillons le problème d'optimisation du routage et de l'ordonnancement des liens que nous traitons, ainsi que les hypothèses sur le fonctionnement des réseaux que nous considérons.

2. Définition du problème et Hypothèses

Le réseau est modélisé par un graphe orienté $G = (V, E)$ de N nœuds représentant les routeurs sans fil qui forment l'infrastructure fixe du réseau. Chaque routeur possède une seule interface lui permettant d'émettre ou recevoir des paquets sur la fréquence radio disponible. Cette fréquence est par conséquent partagée entre tous les routeurs du réseau. L'ensemble des sommets du graphe est décomposé en deux sous-ensembles d'intersection vide : $V = V_r \cup V_g$, avec $V_r \cap V_g = \emptyset$. V_g représente l'ensemble des points d'accès et V_r l'ensemble des routeurs dits simples, c'est-à-dire qui n'intéragissent pas directement avec Internet. Chaque routeur r de V_r possède un poids d_r correspondant à la demande aggrégée des clients mobiles qui se connectent à lui. L'objectif est de router chaque demande d'un routeur source de V_r vers au moins un point d'accès de V_g à travers des chemins multi-sauts.

Le réseau est supposé synchrone et périodique de période T. Le protocole de communication est alors conçu sur un modèle de multiplexage TDMA par intervalles. Ce mécanisme effectue une allocation dynamique de trames à chaque communication pour que tous les usagers puissent accéder à la ressource commune. Une allocation optimale correspond à un ordonnancement optimal des communications de manière à éviter les interférences radio. Cet ordonnancement est assuré par le protocole d'accès au médium (MAC). Afin d'éviter les collisions provoquées par les interférences, la norme WiFi 802.11 se base sur un protocole d'accès CSMA/CA avec messages RTS-CTS qui va autoriser ou refuser la transmission d'un nœud suivant la disponibilité du canal radio. De nombreux modèles d'interférences se basant sur la méthode CSMA/CA ont été développés, chacun d'eux étant une approximation de la réalité physique. Ces modèles sont dits *binaires* : étant donné un graphe représentant la topologie du réseau, nous pouvons représenter un *graphe des conflits* $G_c = (V_c, E_c)$ où chaque nœud de V_c correspond à un arc de E, et il existe un lien entre deux sommets de V_c si et seulement si les deux liens correspondants dans le réseau interfèrent entre eux [KUM 05].

Définissons pour chaque lien e de E l'ensemble $\mathcal{I}(e)$ des liens e' qui interfèrent avec e. Etant donné le modèle d'interférence binaire considéré, s est un ensemble

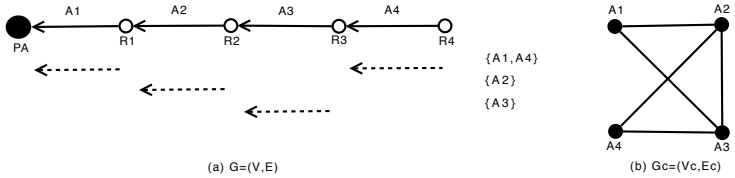


Figure 2. *Modèle binaire d'interférences à distance 2 : (a) Graphe de connectivité, (b) Graphe des conflits. Les ensembles de communications compatibles* $\{A1, A4\}, \{A2\}, \{A3\}$ *constituent des stables du graphe des conflits.*

d'arcs de G n'interférant pas deux à deux : $s \subseteq E$, $\forall e_1, e_2 \in s,\ e_1 \notin \mathcal{I}(e_2)$ et $e_2 \notin \mathcal{I}(e_1)$. Comme E_c relie deux liens qui interfèrent entre eux dans le réseau, tout ensemble s correspond à un ensemble indépendent, ou *stable*, du graphe des conflits G_c. Rappelons qu'un stable est un ensemble de sommets non adjacents deux à deux. Nous illustrons cette notion dans la figure 2 où le modèle d'interférences choisi est un modèle binaire à distance 2, c'est-à-dire que deux liens peuvent être actifs simultanément s'ils se trouvent à une distance au moins égale à 2 dans le graphe. Les arcs $A1$ et $A4$ se trouvent à distance 3 l'un de l'autre, ils peuvent donc se trouver dans un même ensemble s. Nous pouvons vérifier que cet ensemble s est bien un stable du graphe des conflits G_c associé car les sommets $A1$ et $A4$ dans G_c ne sont pas reliés.

Le problème de l'ordonnancement cherche donc à déterminer l'ensemble des liens actifs durant chaque intervalle de temps. Pour qu'une communication puisse avoir lieu d'un routeur à un point d'accès, il faut que les liens portant un chemin de la source à la destination soient activés successivement. Lorsque le réseau fonctionne en régime permanent, périodique, si chaque lien composant un chemin est activé dans la période de longueur T, le flot du chemin aura un débit $\frac{c}{T}$, où c est la capacité du lien de moins grand débit le long du chemin. Le problème du routage de débit maximum consiste donc à trouver un ensemble de chemins entre les sources et les destinations qui soit porté par des liens pouvant être activés dans un intervalle d'amplitude minimale.

3. Optimisations jointes du routage et de l'ordonnancement

Le problème que nous considérons consiste à calculer simultanément l'ensemble des chemins reliant les routeurs aux points d'accès et les transmissions à activer pour écouler le trafic le long de ces chemins.

Dans cette section, nous présentons deux programmes linéaires exacts pour ce problème d'optimisation. Le premier, dit *sommet/arc*, exprime notre problème sur un ensemble de variables associées aux éléments du réseau. Les équations expriment la manière dont ces éléments interagissent. Le second programme, dit *chemin/stable*, se fonde sur une analyse combinatoire plus détaillée du problème. En reportant la structure des solutions dans les variables manipulées, il devient possible de relaxer le problème efficacement et d'envisager des méthodes de résolution sophistiquées.

3.1. *Formulation Sommet/Arc*

Cette formulation du problème joint du routage et de l'ordonnancement, dite en sommet/arc, se base sur les contraintes du problème connu du *multiflot* auquel s'ajoutent les contraintes d'ordonnancement.

Le multiflot donne les chemins entre les routeurs et les points d'accès. Il exprime qu'un routeur r injecte un trafic d_r dans le réseau, que celui s'achemine sans perte de nœud en nœud, puis est extrait aux points d'accès. La capacité de chaque lien e se partage entre les flux f_e^r qui l'empruntent.

L'ordonnancement se décrit par des variables y_t et a_e^t indiquant les intervalles au cours desquels chaque lien est activé. Des contraintes expriment la contention entre deux transmissions interférant entre elles. La longueur de la période du réseau correspond au nombre d'intervalles durant lesquels des liens sont effectivement activés. Une borne sur cette longueur est indiquée, T_{max}, et l'ensemble des intervalles utiles dans la fenêtre $[1, T_{max}]$ est minimisé.

$$\min \sum_{t \leqslant T_{max}} y_t \tag{1}$$

$$\sum_{e \in \Gamma^+(v)} f_e^r - \sum_{e \in \Gamma^-(v)} f_e^r = 0, \forall\, r \in V_r,\ v \in V_r \setminus \{r\} \tag{2}$$

$$\sum_{e \in \Gamma^+(r)} f_e^r = d_r, \forall\, r \in V_r \tag{3}$$

$$\sum_{g \in V_g} \sum_{e \in \Gamma^-(g)} f_e^r = d_r, \forall\, r \in V_r \tag{4}$$

$$a_e^t + a_{e'}^t \leqslant 1,\ \forall e \in E, e' \in \mathcal{I}(e), t \leqslant T_{max} \tag{5}$$

$$\sum_{r \in V_r} f_e^r \leqslant \sum_{t \leqslant T_{max}} a_e^t,\ \forall e \in E \tag{6}$$

$$a_e^t \leqslant y_t,\ \forall e \in E, t \leqslant T_{max} \tag{7}$$

Les contraintes (2)-(4) décrivent le flot des données parcourant le réseau depuis chaque routeur, V_r, qui envoie un trafic d_r aux passerelles V_g.
Les contraintes (5) assurent la compatibilité des transmissions activées durant l'intervalle t : lorsqu'un lien e est activé, tous les liens e' de $\mathcal{I}(e)$ (i.e. interférant avec e) doivent rester inactifs.
Les contraintes (6) lient le routage et l'ordonnancement : la capacité globale d'un lien dépend du nombre d'intervalles durant lesquels il est activé.
Les contraintes (7) comptent le nombre d'intervalles effectivement utilisés durant la période $[1, T_{max}]$, c'est à dire ceux durant lesquels au moins un arc est activé. L'objectif est alors de minimiser le nombre de ces intervalles.

Cette formulation linéaire en nombres entiers possède un nombre important de variables binaires et génère des milliers de contraintes. Elle voit donc sa combinatoire exploser dès que les topologies dépassent une vingtaine de nœuds. Des solutions optimales au problème de routage et d'ordonnancement suivant un objectif dual ont par ailleurs été obtenus sur des petites topologies dans [GOM 07]. Afin de traiter des topologies plus importantes, une autre approche est nécessaire, notamment en relâchant partiellement le problème.

Une relaxation continue de cette formulation en sommet/arc serait sans intérêt car la structure d'un ordonnancement est intimement liée à la binarité des variables a_e^t. Il faut donc reformuler le problème de manière à intégrer la combinatoire dans les variables utilisées et non dans les équations.

On peut par ailleurs remarquer qu'un ordonnancement des transmissions n'est pas indispensable. En effet, une permutation des intervalles laisse le coût de la solution

inchangé. Il suffit donc de connaître la liste des ensembles de transmissions activés dans une solution optimale pour construire un ordonnancement optimal, en affectant à chaque ensemble un intervalle dans un ordre arbitraire. Le problème revient alors à calculer une simple pondération des ensembles de transmissions. En permettant à cette pondération d'être continue et non entière, une relaxation du problème original conserve pour autant sa structure principale.

En se fondant sur ces remarques, nous construisons une nouvelle formulation décrite ci-dessous.

3.2. *Formulation Chemin/Stable*

Comme nous l'avons défini précédemment, un stable correspond à un ensemble de transmissions pouvant être activés simultanément. L'ordonnancement peut alors être considéré comme l'activation d'un stable à chaque intervalle de la période du réseau. Comme expliqué ci-dessus, l'ordre de cette séquence d'activation ne détermine pas la qualité de la solution. Il nous suffit donc de calculer l'ensemble des stables activés. De la même manière, nous pouvons caractériser le routage comme la sélection d'un ensemble de chemins sur lesquels le flot va s'écouler.

Nous allons maintenir décrire le problème de routage et d'ordonnancement à partir de ces objets. Ainsi la structure combinatoire d'une solution est donnée par les variables et non plus par les contraintes. Il sera donc possible d'en faire une relaxation continue qui soit efficace.

Chaque stable s peut être activé un certain nombre de fois a_s durant la période du réseau ; lorsque le problème est relâché, a_s représente le temps durant lequel s est activé. Un seul stable pouvant être activé à la fois, la somme des temps d'activation des stables doit être égale à la longueur de la période du réseau, celle là même que l'on cherche à minimiser. La capacité d'un lien est alors donnée par la somme des temps d'activation des stables dont il fait partie.

Etant donné l'ensemble des chemins $\mathcal{P}_r$ allant du nœud r vers les passerelles, $f_p, p \in P_r$ correspond à la quantité de flot envoyée sur le chemin p depuis r jusqu'à la passerelle située à son extrémité.

Dans la suite, nous notons $\mathcal{P} = \cup_r \mathcal{P}_r$, l'ensemble des chemins utilisés, et $\mathcal{S}$ l'ensemble des stables. Nous pouvons alors formuler le problème d'ordonnancement et de routage de la façon suivante.

$$\min \sum_{s \in \mathcal{S}} a_s \tag{8}$$

$$\sum_{p \ni e} f_p \quad \leqslant \quad \sum_{s \ni e} a_s, \ \forall e \in E \tag{9}$$

$$\sum_{p \in \mathcal{P}_r} f_p \quad = \quad d_r, \ \forall r \in V_r \tag{10}$$

Le nombre de contraintes de cette formulation est fortement réduit puisque les contraintes de conservation de flot et de compatibilité des transmissions n'ont plus lieu d'être. Seuls le respect des capacités par le flot (eq. 9) et l'envoi du trafic dans le réseau (eq. 10) doivent être exprimés.

En contrepartie, le nombre de variables explose. En effet, le nombre de chemins entre deux points dans un graphe, ainsi que le nombre de stables sont exponentiels. Il n'est donc pas possible de manipuler concrètement cette formulation avec l'ensemble complet des variables.

En particulier, la technique de génération de colonnes permet d'éviter l'énumération de la totalité des variables, comme décrit ci-après.

4. Génération de Colonnes

La génération de colonnes permet de manipuler les programmes linéaires ayant un nombre exponentiel de variables.

Il s'agit alors de résoudre la relaxation continue de la formulation *chemin/stable* sur des ensembles de chemins et de stables, $\mathcal{P}_0$ et $\mathcal{S}_0$, restreints. Ce programme, appelé alors *programme maître*, se résout très rapidement. Il suffit ensuite d'étudier son dual. Les variables de celui-ci forment une pondération sur les arcs et les sommets du graphe. A partir de celles-ci, il est possible de calculer des chemins et des stables permettant d'améliorer la solution actuelle du maître. Ces objets, générés par des programmes dits *auxiliaires*, forment des nouvelles *colonnes* du maître. Ce processus est itéré jusqu'à ce que les programmes auxiliaires ne trouvent plus de colonnes améliorantes.

4.1. *Programme dual*

Le programme dual associé au programme maître décrit ci-dessus possède deux contraintes correspondant aux variables f_p, et a_s. Soient $y_e,\ e \in E$ les variables duales associées à la contrainte (9), et $x_u,\ u \in V_r$ associées à la contrainte (10). Le programme dual consiste alors à maximiser une forme de "*volume*" du graphe, $\sum_{r \in V_r} d_r x_r$, sous les contraintes de longueur de chemin et de poids de stable qui suivent.

$$\sum_{e \in p} y_e \quad \geqslant \quad x_{\mathcal{O}(p)},\ \forall p \in \mathcal{P} \tag{11}$$

$$1 \quad \geqslant \quad \sum_{e \in s} y_e,\ \forall s \in \mathcal{S} \tag{12}$$

où $\mathcal{O}(p) \in V$ correspond au nœud source du chemin p.

Le problème est maintenant de déterminer s'il existe des chemins et des stables qui violent les contraintes du programme dual et améliorent la valeur de la fonction objective du programme maître.

4.2. *Problèmes auxiliaires*

La résolution par génération de colonnes du problème du routage et de l'ordonnancement que nous présentons entraîne la définition de deux problèmes auxiliaires.

Le premier consiste à trouver un *chemin pondéré* par les variables duales associées aux arcs violant la contrainte du dual, c'est à dire de poids inférieur à la variable duale associée à sa source (cf equation 11). Les plus courts chemins des nœuds vers les points d'accès permettent donc toujours de savoir si une contrainte n'est pas violée, ou de fournir un bon candidat. Le cas échéant, ce chemin est ajouté à l'ensemble courant des variables du programme maître, car il permet d'en améliorer l'objectif.

De même, étant donnée cette pondération, le deuxième problème auxiliaire consiste à calculer un ensemble de communications compatibles de poids supérieur à 1 (cf equation 12). Dans notre cas, le stable de poids maximum du graphe des conflits est le séparateur adéquat. Il est à noter que calculer un stable est NP-difficile sur des graphes généraux. Nous verrons par la suite qu'il ne s'agit pas là d'un enjeu majeur de complexité pratique.

5. Simulations

Le processus de génération de colonnes a été implémenté en utilisant la bibliothèque MASCOPT (`http://www-sop.inria.fr/mascotte/mascopt/`) et l'API du solveur ILOG CPLEX.

Des tests ont été réalisés sur un ensemble d'instances aléatoires. Un ensemble de n points est généré sur une bande de longueur 1 et de hauteur $\frac{1}{4}$ selon un processus de Poisson. Un rayon de communication est ensuite calculé de sorte à ce que le réseau soit connexe et de degré moyen $\bar{d}$ proche de $\max\left(5, \frac{n}{10}\right)$. Les passerelles sont choisies aléatoirement et uniformément parmi les points. Les autres points sont des routeurs.

Nous avons généré des topologies de taille allant de 10 à 100. Pour chaque topologie, nous avons lancé des tests pour un nombre de passerelles allant de 1 à $\frac{n}{\bar{d}}$. Dans ce dernier cas, les passerelles ne communiquent en moyenne qu'avec leurs voisins.

En renormalisant les distances entre les nœuds de telle sorte que le rayon de communication soit 1, on remarque que ce processus génère des graphes poissonniens d'intensité $\frac{n}{10\pi}$. Les graphes sont donc localement denses, mais, du fait de la forme rectangulaire de la zone considérée, ils restent étalés.

Nous présentons dans la suite les résultats obtenus sur la capacité du réseau et des intuitions qui en découlent. Nous discutons notamment des questions de complexité.

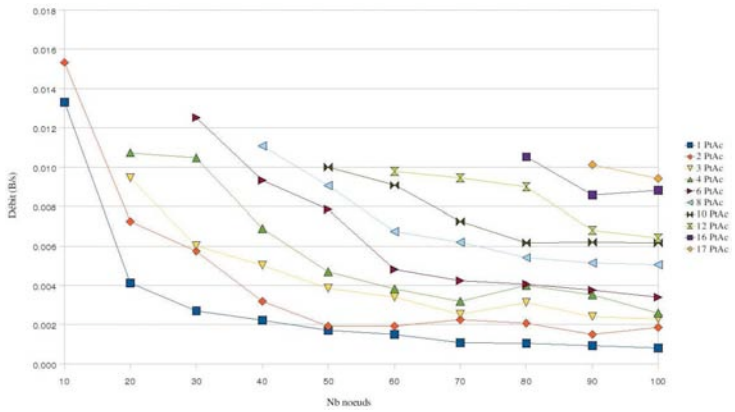


Figure 3. *Évolution du débit avec la taille du réseau (1 à 19 points d'accès)*

5.1. *Débit et taille du réseau*

Les travaux sur la capacité des réseaux lors d'une collecte de données par une passerelle indiquent que le débit disponible à chaque routeur devrait décroître proportionnellement à l'inverse de la taille du réseau [KLA 04]. L'intuition naturelle est que le même phénomène apparaît lorsque plusieurs passerelles sont présentes.

La figure 3 confirme cette intuition. Une analyse détaillée du problème devrait montrer qu'en dehors d'une zone entourant chaque passerelle où se concentrent l'essentiel des contentions, le programme linéaire n'est pas fortement contraint.

Il deviendrait alors possible d'envisager des algorithmes bien plus efficaces se contentant d'optimiser autour des passerelles. Les zones périphériques peuvent alors être traitées par des processus rapides et peu précis (par exemple gloutons). Cette intuition est appuyée par une complexité pratique de calcul des stables sous-linéaire avec la taille des réseaux, mais sensible au nombre de passerelles (et donc à la taille des zones de contention maximale).

5.2. *Débit et densité de passerelles*

La dépendance du débit à la densité des passerelles est moins claire a priori. Une dépendance linéaire est une borne supérieure évidente, mais les interférences entre passerelles pourraient dégrader les performances. Pour étudier ce phénomène, nous avons généré, pour chaque topologie, plusieurs tirages aléatoires du positionnement des points d'accès. La figure 4 présente la moyenne du gain obtenu en ajoutant des passerelles. Les valeurs sont normalisées par le débit avec une passerelle.

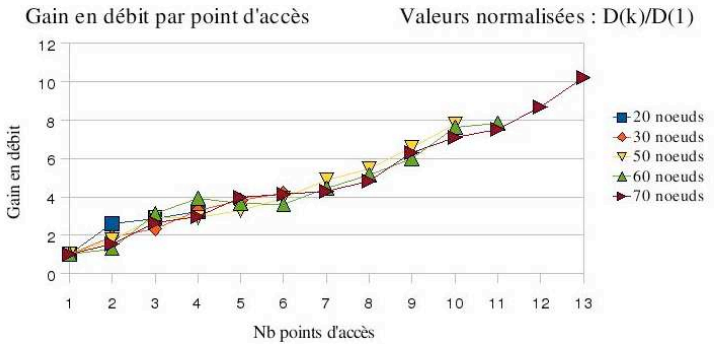


Figure 4. *Gain en débit par point d'accès (20 à 70 nœuds)*

Ces résultats montrent un gain linéaire en fonction du nombre de passerelles, avec une pente légèrement inférieure à 1. Lorsque le placement est correct, même non optimisé, les réseaux maillés se comportent efficacement. Une analyse plus détaillée des résultats en fonction de la répartition des passerelles dans le réseau montre que leur placement a un impact très important. Lorsque deux passerelles sont trop proches l'une de l'autre, elles se comportent comme une seule. En effet, les interférences radio les empêchent de collecter le trafic simultanément.

Nous conjecturons qu'une distance minimale autour de chaque passerelle est non seulement nécessaire mais aussi suffisante à un écoulement correct du trafic. Cette distance devrait pouvoir se calculer explicitement sur des topologies régulières.

6. Conclusion

Dans cet article, nous avons modélisé l'optimisation jointe du routage et de l'ordonnancement des transmissions dans les réseaux radio maillés dédiés à l'accès à Internet. Nous avons développé des formulations exactes en programmation linéaire ainsi qu'un processus de génération de colonnes. Cette approche permet de calculer des solutions sur des réseaux de grande taille et d'étudier le comportement de la capacité offerte aux clients suivant différents paramètres.

Nos expérimentations mettent en évidence l'impact du positionnement des points d'accès dans le réseau qui doit être considéré avec attention. Il semble cependant qu'il ne soit pas indispensable de l'optimiser finement. Une étude combinatoire devrait permettre de montrer l'existence d'une zone sensible centrée sur les passerelles au-delà de laquelle le problème peut se résoudre grossièrement.

Ces pistes peuvent avoir autant de conséquences du point de vue de la capacité des réseaux maillés que de la complexité de leur conception.

7. Bibliographie

[AKY 05] AKYILDIZ I., WANG X., WANG W., « Wireless mesh networks : a survey », *Computer Networks*, vol. 47, n° 4, 2005, p. 445-487.

[CAR 07] CARELLO G., FILIPPINI I., GUALANDI S., MALUCELLI F., « Scheduling and routing in wireless multi-hop networks by column generation », *INOC 2007*, 2007.

[DOU 04] DOUSSE O., FRANCESCHETTI M., TSE D., THIRAN P., « Closing the gap in the capacity of random wireless networks », *IEEE International Symposium on Information Theory (ISIT)*, 2004.

[GOM 07] GOMES C., RIVANO H., « Fair Joint Routing and Scheduling Problem in Wireless Mesh Networks », Research Report n° 6198, 2007, INRIA.

[GUP 00] GUPTA P., KUMAR P., « The Capacity of Wireless Networks », *IEEE Transactions on Information Theory*, vol. 46, n° 2, 2000, p. 388-404.

[JAI 03] JAIN K., PADHYE J., PADHAMANABHAN V., QIU L., « Impact of Interference on Multi-hop Wireless Network Performance », *ACM MobiCom*, 2003, p. 66-80.

[JUN 03] JUN J., SICHITIU M., « The nominal capacity of wireless mesh networks », *IEEE Wireless Communications*, vol. 10, n° 5, 2003, p. 8-14.

[KLA 04] KLASING R., MORALES N., PERENNES S., « On the Complexity of Bandwidth Allocation in Radio Networks with Steady Traffic Demands », Research report, 2004, INRIA/RR-5432 and I3S/RR-2004-40-FR.

[KOD 05] KODIALAM M., NANDAGOPAL T., « On the capacity region of multi-radio multi-channel wireless mesh networks », *First IEEE WiMesh*, 2005.

[KUM 05] KUMAR A., MARATHE M., PARTHASARATHY S., SRINIVASAN A., « Algorithmic aspects of capacity in wireless networks », *ACM SIGMETRICS*, vol. 33, 2005, p. 133-144.

[LAK 06] LAKSHMANAN S., SUNDARESAN K., SIVAKUMAR R., « On Multi-Gateway Association in Wireless Mesh Networks », *IEEE WiMesh*, 2006, p. 135-137.

[LIU 05] LIU H., ZHAO B., « Optimal Scheduling for Link Assignment in Traffic-Sensitive STDMA Wireless Ad-Hoc Networks », *3rd International Conference on Networking and Mobile Computing (ICCNMC)*, vol. 3619 de *Lecture Notes in Computer Science*, 2005, p. 218-228.

[LUB 05] LUBBECKE M. E., DESROSIERS J., « Selected Topics in Column Generation », *Operations Research*, vol. 53, n° 6, 2005, p. 1007-1023.

[MEH 06] MEHEUT G., PERENNES S., RIVANO H., « Evaluation stochastique et simulation des réseaux radio », Research report n° 5989, 2006, INRIA.

[RIV 06] RIVANO H., THEOLEYRE F., VALOIS F., « Capacity Evaluation Framework and Validation of Self-Organized Routing Schemes », *IEEE International Workshop on Wireless Ad-hoc and Sensor Networks (IWWAN)*, 2006.

[TAM 07] TAM W.-H., TSENG Y.-C., « Joint Multi-Channel Link Layer and Multi-Path Routing Design for Wireless Mesh Networks », *IEEE INFOCOM*, 2007.

[WAN 07] WANG P., BOHACEK S., « Toward tractable computation of the capacity of multi-hop wireless networks », *IEEE INFOCOM*, 2007.

[ZHA 05] ZHANG J., WU H., ZHANG Q., LI B., « Joint Routing and Scheduling in Multi-radio Multi-channel Multi-hop Wireless Networks », *IEEE BROADNETS*, 2005, p. 678-687.

Design d'une interface cross-layer et implémentation de métriques complexes pour les réseaux ad-hoc

Hervé Aïache, Vania Conan, Laure Lebrun, Jérémie Leguay, Stéphane Rousseau, Damien Thoumin

Thales Communications[1]
160 Bd de Valmy, BP 82
92704 Colombes cedex, France
{firstname.name}@fr.thalesgroup.com

RÉSUMÉ. *L'approche* cross-layer *suscite de plus en plus d'intéréts dans le domaine des réseaux mobiles. Toutefois, les outils de validation expérimentale sont encore peu nombreux. Ce papier présente une extension logicielle majeure à XIAN (Cross-layer Interface for wireless Ad hoc Networks). Il repose sur un framework générique permettant d'expérimenter l'implémentation de design* cross-layer *sous plateforme Linux avec des cartes sans fil 802.11 en utilisant le driver MadWifi. XIAN rend accessible les informations disponibles au niveau des couches MAC/PHY et offre un cadre expérimental permettant la création de métriques pouvant utiliser des informations locales et/ou des informations de voisinages. Afin d'illustrer les possibilités données par cette extension, nous prenons comme exemple l'implémentation de la métrique ETX (Expected Transmission Count) et fournissons les résultats de cette étude expérimentale.*

ABSTRACT. *In the highly dynamic and unpredictable environment of MANETs,* cross-layer *design is receiving growing interest but lacks experimental validation tools. This paper presents XIAN (Cross-layer Interface for wireless Ad hoc Networks), a generic framework for experimenting cross-layer designs in Linux testbeds with 802.11 wireless cards using the MadWifi driver. XIAN can be used as a service by other layers to access MAC/PHY configuration and performance information and to create automatically complex metrics from both local and neighbour node measurements. The defined and implemented software architecture introduces the XIAN Nano-protocol and its automated management. We exemplify their benefits through the implementation of the well-known ETX (Expected Transmission count) metric and we provide results from real experimentations.*

MOTS-CLÉS : *Réseaux Ad hoc, Cross-layer, QoS, métrique*

KEYWORDS: *Ad hoc Networks, Cross-layer, QoS, metric*

1. Ce travail a été soutenu par le projet RNRT Airnet et le projet européen IST FP6 Chorist

1. Introduction

Outre le fait que dans les Réseaux Ad hoc Mobiles (MANETs [COR 99]), les nœuds sont susceptibles de se déplacer, la qualité des liens radio établis entre deux noeuds évolue au cours du temps. En effet, la connexion radio peut passer d'un état très bon à un état très faible en très peu de temps et inversement. Certaines études montrent que le trafic du réseau lui-même est à l'origine de ces dégradations.

D'autre part, on souhaite accueillir dans les réseaux ad hoc de plus en plus d'applications nécessitant une certaine qualité de service. Il est donc nécessaire d'avoir une connaissance très précise de l'état du réseau à un instant donné afin de pouvoir satisfaire ce genre d'applications. Un des objectifs de notre étude est de trouver un compromis entre connaître l'état du réseau et ne pas surcharger inutilement les liens radio par des paquets de contrôle. Ainsi, la bande-passante disponible pour les applications n'est pas diminuée par d'innombrable paquets de contrôle.

Le modèle OSI tel qu'il est présenté permet de déceler l'évolution de la qualité des liens grâce à l'envoi fréquent de paquets de contrôle. Cet envoi est fait non seulement par la couche MAC mais aussi par les couches supérieures telles que la couche réseau. Ce mode de fonctionnement engendre ainsi des duplications d'information et de contrôle au sein d'un même noeud et par conséquent engendre une perte de bande-passante.

Notre approche consiste à réutiliser les informations disponibles dans les couches basses et de les remonter vers les couches supérieures. Cette approche, connue sous le nom cross-layer, permet de diminuer considérablement le nombre de paquets de contrôle nécessaires pour vérifier la qualité des liens (suivant différents critères). La communication peut se faire soit entre les couches adjacentes (e.g. Kawadia et al. [KAW 03]), ou éventuellement entre des couches non adjacentes (e.g. Conti et al. [CON 04]). Si cette approche existe déjà, son expérimentation reste difficile du fait d'un manque de support API approprié des cartes 802.11 utilisées dans les testbeds ad hoc.

Ce papier propose une extension de XIAN [AIA 06] (Cross-layer Interface for wireless Ad hoc Networks) qui rend accessible des informations contenues dans les couches MAC/PHY et facilite l'intégration et l'évaluation de concepts *cross-layer*. XIAN est aujourd'hui disponible sur Linux avec le driver *MAD* [MAD] 802.11. XIAN a été conçu d'une part dans le but d'encourager les implémentations basées sur le cross-layer, et de faciliter son déploiement dans les plateformes MANET. L'extension apporte la possibilité d'ajouter de nouvelles métriques plus ou moins complexes utilisant des informations locales ou provenant des noeuds voisins. Le *nano-protocole XIAN* a pour vocation de faciliter l'ajout de ces métriques. Nous illustrons ces nouvelles fonctionnalités à la fin du papier, par l'implémentation de la métrique ETX proposée par De Couto et al. [DEC 03].

L'organisation du papier est la suivante. Dans la partie 2, nous présentons l'approche et le concept donnés dans XIAN. La partie 3 décrit le *nano-protocole XIAN*.

Les parties 4 et 5 sont consacrées à l'implémentation et l'application de la métrique ETX sur une étude expérimentale. Enfin la partie 6 conclut le papier et donne différentes directions pour les prochaines évolutions de XIAN et des approches cross-layer.

2. XIAN : une interface Cross-layer pour les réseaux sans fil ad hoc

XIAN a été développé dans le but de faciliter les études expérimentales dans les réseaux Ad-Hoc. Pour cela, XIAN simplifie considérablement les communications entre les couches adjacentes mais aussi non-adjacentes du modèle OSI. La plupart des informations échangées sont des métriques permettant de statuer sur la qualité des liens et des ressources disponibles. Ces données calculées par les couches basses sont immédiatement utilisables par les couches supérieures et peuvent être réutilisées pour calculer de nouvelles métriques. Ces échanges se font par un mécanisme de question/réponse entre les couches. Nous décrivons ici en détails l'architecture logicielle de XIAN et nous présentons le type d'informations/états/statistiques qu'il offre aux autres composants du système Linux.

2.1. *Architecture logicielle de XIAN*

XIAN contient trois composants principaux.

– Le ***Kernel Space XIAN Interface*** (KSI) est implémenté comme un module noyau. Il interagit directement avec le driver *MadWifi* pour en extraire les statistiques et les états de chaque connexion.

– Le ***User Space XIAN Interface*** (USI) duplique le KSI mais au niveau de l'espace utilisateur. Cette API est contenue dans une librairie C dans le but de faciliter son intégration au sein de programmes utilisateur. (e. g. les démons de routage et les applications).

– Le ***XIAN Information Transport Module*** (ITM) permet de récupérer des informations et des statistiques contenues dans le driver *MadWifi* en passant par les Interfaces décrites précédemment. Ce module est implémenté dans cette version de XIAN comme un *"character device"*.

En sus, un composant complémentaire, le ***XIAN User Space Extended Interface*** (USEI), fournit aux utilisateurs un moyen simple pour faire des moyennes, des combinaisons de métriques ou encore noter des changements significatifs pour une métrique particulière. Il intéragit directement avec le USI et peut être décomposé en trois ensembles de fonctions principaux :

– ***Measurement functions***, qui fournit des métriques calculées à partir de métriques élémentaires prises directement via l'USI ou à partir de moyennes.

– ***Operation functions***, où sont implémentés les opérateurs mathématiques nécessaires au calcul de certaines métriques (telles que la moyenne, le min, le max, ...).

– ***Relevance functions***, qui met en oeuvre le comparateur correspondant permettant d'indiquer si une différence significative arrive entre deux métriques calculées (typiquement entre les précédents et les nouveaux).

Ces cinq composants sont représentés dans la figure 1. De plus, nous y indiquons la manière dont ils interagissent les uns avec les autres et comment on fournit les états internes du driver MAC aux autres composants Linux.

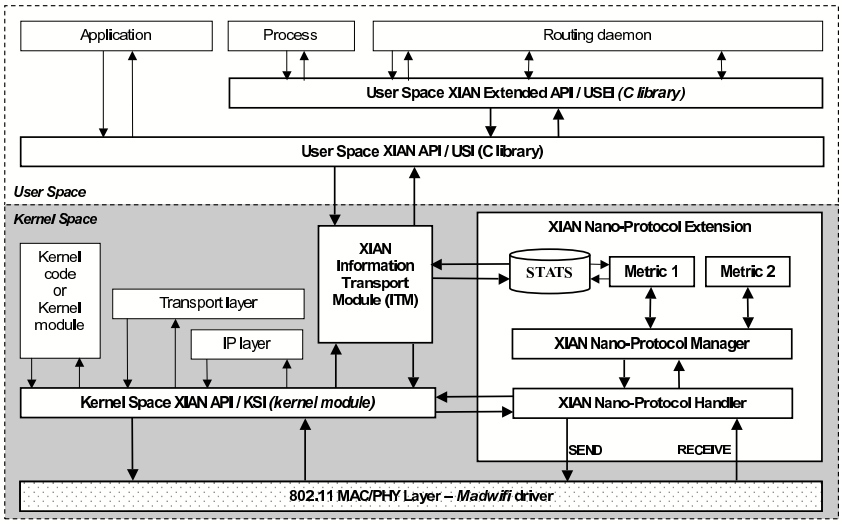


Figure 1. *Architecture logicielle de XIAN pour linux.*

L'ITM est un module orienté message implémenté comme un *character device*. Les métriques échangées via son interface, rendues accessibles à USI et KSI, sont de deux types :

– des métriques de base maintenues à jour par le driver *MadWifi*

– des métriques complexes implémentées dans l'extension *XIAN nano-protocol* présentée dans ce papier.

2.2. *Les interfaces de base de XIAN*

XIAN permet un accès facile à différentes métriques de base disponibles dans le driver *MadWifi*. Elles peuvent être organisées en trois groupes :

– ***Etats de configuration***, qui concerne la configuration courante des différents paramètres du périphérique 802.11. On peut par exemple y récupérer la taille des files

d'attente et la fréquence utilisée.

– ***Aggregated metrics***, il s'agit de compteurs qui fournissent le statut global d'utilisation de l'interface 802.11. Par exemple, les informations rapportées peuvent être le nombre de trames reçues, rejetées ou avec un mauvais BSSID, le nombre de réceptions échouées (e.g. à cause d'un mauvais CRC ou d'un problème de décryptage).

– ***Per neighbour/link metrics***, qui conserve les informations par voisin récupérées par la couche MAC, par exemple le nombre de trames de données ou bytes reçues/transmises, la force du signal (RSSI) ou le nombre de transmissions renouvelées.

Ainsi, environ 180 fonctions ont été développées et intégrées dans l'API XIAN afin de couvrir chacune des métriques disponibles (i.e. dans KSI et dans USI, la duplication de l'interface au niveau de l'utilisateur. Sur ces 180 métriques, 40 d'entre elles contiennent des informations du voisinage et le reste concerne plus particulièrement le noeud lui-même.

3. L'extension *XIAN nano-protocol*

Comme indiqué sur la figure 1, l'extension *XIAN nano-protocol*, disponible sur le site officiel [XIA], offre des mécanismes implémentant des échanges bidirectionnels de métriques entre noeuds voisins. Le format de ces messages est donné sur la figure 2. Ces paquets sont envoyés de manière périodique et contiennent les informations sur les mesures cross-layer définies. Une fois reçues, ces métriques sont collectées par le Kernel Space XIAN Interface, qui interagit directement avec le driver MadWifi.

A chaque métrique est associé un nombre de paramètres de configuration, par exemple la fréquence à laquelle sont envoyés les paquets sur le réseau, la fonction de traitement invoquée à la réception de ces paquets pour mettre à jour la métrique. Une configuration type pour une métrique se caractérise par : un identifiant de configuration, un nombre associé à la métrique, le périphérique à utiliser, l'adresse MAC destination pour le rapport, la fréquence d'envoi (en millisecondes) et un pointeur de fonction qui se charge du traitement de l'information. Chaque métrique est implémentée sous la forme d'un module noyau Linux utilisant l'API fournie par XIAN.

Dans le protocole que nous avons implémenté, l'élément principal du message XIAN est la métrique. Les caractéristiques d'une métrique XIAN sont son type (i.e. un nombre identifiant la métrique), un identifiant de la configuration de la métrique, un type d'encodage, l'adresse MAC associée à la métrique et la valeur de la métrique. Les autres champs composent l'entête du message XIAN :

– ***Version*** : la version du nano-protocole XIAN.

– ***Id*** : l'identifiant du message.

– ***Length*** : le nombre de métriques composant le message.

– ***Payload*** : un ensemble d'informations sur les métriques.

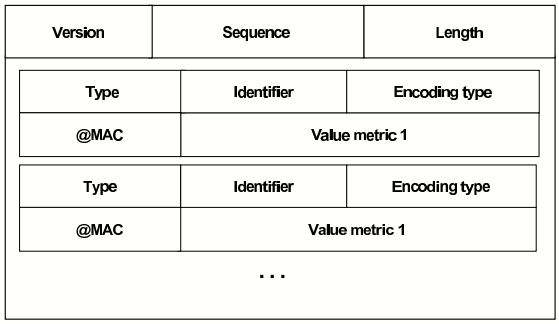


Figure 2. *Format des messages du nano-protocole XIAN*

Comme montré sur la figure 1, quatre composants logiciels, tous implémentés en tant que module noyau, supporte l'intégration de métriques complexes au sein du framework XIAN.

3.1. *Le XIAN Nano-Protocol Handler*

Ce module se charge des messages XIAN échangés entre les noeuds. Sa première fonctionnalité est d'envoyer les messages XIAN reçus vers le module XIAN Nano-Protocol Manager, qui se trouve juste au-dessus dans l'architecture.

De plus, ce module se charge de construire et envoyer les messages XIAN requis par les métriques complexes ayant été implementées dans le framework. Diverses métriques sont alors échangées entre les noeuds voisins. Il faut noter que un mécanisme a été mis en place afin de ne pas utiliser la bande-passante de manière excessive lorsque plusieurs métriques complexes requièrent l'échange de même métrique de base.

3.2. *XIAN Nano-Protocol Manager*

Ce module permet la gestion et l'utilisation de plusieurs métriques. Il est au coeur de l'architecture XIAN. Il permet en outre un mécanisme d'abonnement et désabonnement pour la configuration de plusieurs métriques.

A la réception de messages XIAN, le XIAN Nano-Protocol Manager analyse et isole les différentes métriques présentes dans le message. Pour chaque métrique, il exécute la fonction appropriée qui se charge de la mise à jour des valeurs correspondantes. Il a aussi la possibilité de connaître l'adresse MAC du noeud origine du message et si nécessaire lui communiquer des informations. Dans le cas où la métrique n'est pas définie elle est automatiquement ignorée.

3.3. *Le module de statistiques*

Le module précédent est nécessaire pour la mise en place de nouvelles métriques utilisant des informations bi-directionnelles. Une fois traitées, ces métriques doivent être accessibles à d'autres modules ou processus. Cette fonctionnalité est assurée par le module XIAN Statistics. Ce module permet la sauvegarde des différentes métriques dans une base de données. Pour chaque métrique, il est possible de retenir la valeur, l'identifiant de configuration et l'adresse MAC du noeud origine (le noeud qui a envoyé l'information).

3.4. *Le module de configuration de métriques* cross-layer

La fonction principale de ce module est de configurer le comportement pour une métrique donnée. Il est possible de charger plusieurs modules et ainsi définir plusieurs métriques. Plusieurs paramètres peuvent être précisés tels que l'identifiant, le type de métrique, le périphérique utilisé, l'adresse MAC destination, la fréquence (en millisecondes) des rapports et plus particulièrement le traitement souhaité pour cette métrique. Le traitement des métriques peut être soit local soit nécessiter une communication avec les autres noeuds et sera fournie par l'intermédiaire d'un pointeur vers la fonction de traitement.

4. Implémentation de la métrique ETX dans XIAN

Nous allons à présent illustrer tout le fonctionnement de XIAN à travers un exemple simple d'implémentation de métrique. Pour se faire, nous avons choisi la métrique ETX proposée par De Couto et al. [DEC 03]. Cette métrique est bi-directionnelle et permet de connaître la qualité d'un lien et avertir d'une congestion ou d'interférences qui pourraient compromettre la communication. Il est évident que cette métrique peut s'avérer être très utile pour la mise en place de qualité de service dans un réseau. D'autres métriques auraient pu faire l'objet de cette même étude telles que la *médium time metric* (MTM) proposée par Awerbuch et al. [AWE 04] qui sélectionne le meilleur chemin en termes de bande-passante ou bien la *available bandwidth* introduite par Déziel et al. [DEZ 05], ou encore la métrique présentée par Iannone et al. [IAN 04] qui combine le taux de succès de transmission, le niveau d'interférence et le débit. La partie suivante se focalise sur la métrique ETX et son implémentation avec l'extension XIAN nano-protocol.

4.1. *ETX*

La métrique *expected transmission count* (ETX) mesure le nombre moyen de transmissions nécessaires pour la bonne réception d'un paquet. Cette mesure peut être faite sur chaque lien du réseau. Ainsi, pour connaître le nombre moyen de transmis-

sions nécessaires pour atteindre une destination, il suffit de faire la somme d'ETX sur chacun des liens traversés.

ETX est une mesure bi-directionnelle. Elle combine le *forward delivery ratio*, noté Df, et le *reverse delivery ratio*, Dr. Df est la probabilité qu'un paquet arrive avec succès au noeud suivant et Dr est la probabilité que le paquet ACK arrive lui-aussi au noeud source. Une bonne transmission est l'association de ces deux étapes. ETX est donc égale à l'inverse du produit $Df * Dr$.

$$ETX(link) = \frac{1}{Df * Dr} \quad [1]$$

Cette métrique se calcule par l'envoi périodique de paquets aux noeuds voisins. Ces paquets sont comptabilisés et numérotés ce qui permet d'en déduire aisément le nombre de transmissions réussies. Chacun des paquets envoyés contient ce nombre (calculé localement) ce qui permet a un noeud voisin de calculer la valeur d'ETX.

4.2. *Détails de l'implémentation*

La métrique ETX a été implémentée dans le framework XIAN comme un module noyau spécifique. Au moment de charger le module ETX, nous configurons le XIAN Nano-Protocol Manager afin qu'il envoie la valeur Dr dans un message de broadcast niveau MAC toutes les T millisecondes. Un nouveau type ainsi qu'un nouvel identifiant de configuration de la métrique ETX sont alors créés.

Le paramètre pf fourni au moment de l'enregistrement de la métrique via la fonction $register_id$ correspond à la fonction de traitement appelée à chaque réception de la métrique ETX. Le message de broadcast contient la valeur Dr pour la liste d'adresses MAC fournie par la fonction $insert_mac_to_broadcast$ qui est invoquée pour chaque message de broadcast reçu quand aucune adresse MAC n'est présente. A la réception de ces messages, le module noyau correspondant à ETX stocke la valeur de la métrique D_r dans le module de statistiques via leur interface commune et la fonction $update_xian_stat$. Alors le module métrique incrémente un compteur qu'il stocke dans le module de statistiques. A partir des D_r et D_f obtenus, un calcul de ETX est réalisé puis stocké dans le module statistiques mettant ainsi à disposition une estimation de la qualité du lien.

En parallèle, un thread est réveillé toutes les W millisecondes. Il calcule alors D_f avec la valeur du compteur qu'il récupère et le nombre de messages estimés durant ces W millisecondes. Par la suite, D_f est stocké dans le module STATS et le compteur est remis à zéro. ETX est alors recalculé et mis à jour dans le module STATS.

Notons que, comme le noyau Linux ne prend pas en compte des flottants, toutes les métriques dans l'espace noyau sont sous le format spécifique $xian_float$ et accessible par la fonction $struct\ xian_float\ get_xian_stat$ dans le KSI. Dans l'espace utilisateur, les résultats sont disponibles sous le format flottant avec une fonction similaire $float\ get_xian_float()$ dans le USI. Par ailleurs, pour terminer le processus,

la configuration de la métrique doit être effacée via la fonction $unregister_id$ et le module déchargé.

5. Résultats expérimentaux

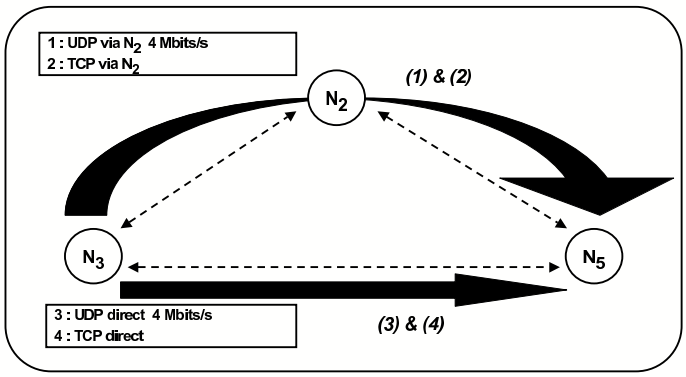
Nous présentons ici les résultats concernant ETX obtenus en plateforme. Ces résultats ont pour but de mettre en évidence les possibilités de XIAN mais aussi de mettre en évidence l'intérêt d'ETX dans le cadre du routage. La plateforme est composée de trois machines équipées de cartes Cisco Aironet Wi-Fi utilisant le chipset Atheros. On utilise le standard 802.11b à 11 Mbits/s en mode ad-hoc (sans RTS/CTS). Sur chaque machine nous utilisons *iperf* [IPE] pour générer du trafic TCP et UDP. De plus, le framework STAF/STAX [STA] permet l'automatisation des expérimentations.

Le mesures de performances suivantes ont été réalisées toutes les $delta$ secondes :

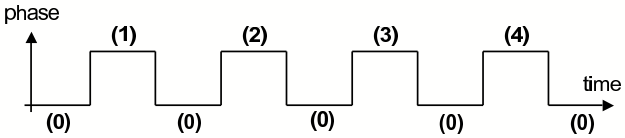
– Le ***RSSI*** (Relative Signal Strength Indicator) : la puissance reçue par les noeuds.

– Le ***Throughput*** : le nombre total d'octets reçus et envoyés par la couche MAC durant les $delta$ dernières secondes.

– L' ***ETX*** calculé sur 10 messages ayant été envoyés toutes les 200 ms.



(a) Topologie du réseau



(b) Chronogramme pour le génération de trafic

Figure 3. *Paramètres de l'expérimentation.*

Les figures 3(a) et 3(b) présentent la topologie du réseau et précise les instants où les flots transitent. Le lien entre les machines N_3 et N_5 est plus grand que les deux autres et ce de manière significative. Le trafic est généré en 5 phases durant chacune 24 secondes et séparées par une seconde.

– ***Phase (0)*** : aucun trafic.

– ***Phase (1)*** : Trafic UDP entre N_3 et N_5 en passant par N_2 avec un débit de 4 Mbits/s.

– ***Phase (2)*** : Trafic TCP entre N_3 et N_5 en passant par N_2.

– ***Phase (3)*** : Trafic UDP entre N_3 et N_5 avec un débit de 4 Mbits/s.

– ***Phase (4)*** : Trafic TCP entre N_3 et N_5.

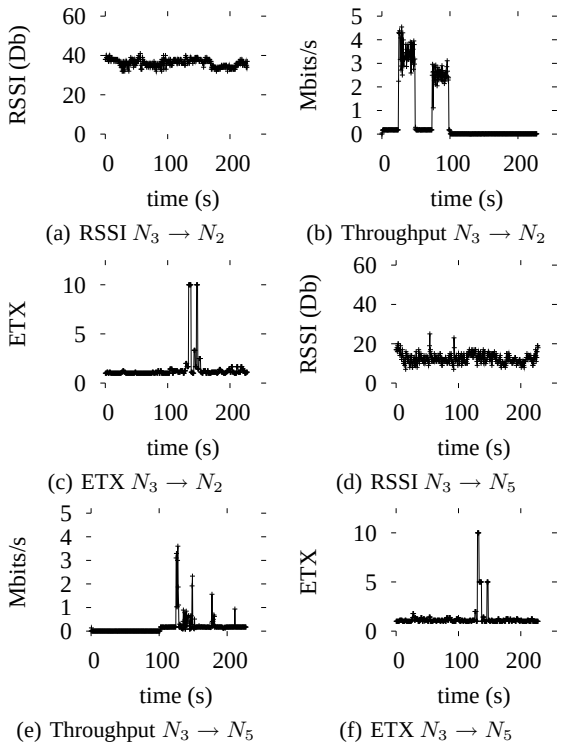
Nous présentons les résultats obtenus expérimentalement. Ces résultats sont regroupés sur la figure 4. Plus précisément, les figures 4(b) et 4(e) indiquent les débits respectifs des liens $N_3 \rightarrow N_2$ et $N_3 \rightarrow N_5$.

Une première analyse nous permet d'estimer la valeur de la bande passante disponible entre N_3 et N_2 et entre N_2 et N_5. En effet, lorsque les flux UDP et TCP issus du noeud N_3 passent par N_2 pour atteindre N_5 nous parvenons à saturer le lien. Nous en déduisons une bande-passante de 3.4 Mbits/s pour le lien $N_3 \rightarrow N_2$ et de 2.4 Mbits/s pour le lien $N_2 \rightarrow N_5$. Au contraire, quand les trafics UDP et TCP issus du noeud N_3 passent directement par le lien $N_3 \rightarrow N_5$, la bande-passante disponible sur les mêmes liens est alors de 0.1 Mbits/s. Une première conclusion est qu'une décision de routage ne tenant compte que du nombre de sauts serait inappropriée dans ce cas. Pour améliorer la qualité des décisions de routage nous introduisons donc le concept cross-layer.

Sur les figures 4(a) et 4(d) nous nous intéressons à la valeur du RSSI pour les liens $N_3 \rightarrow N_2$ et $N_3 \rightarrow N_5$. La distance significative entre les noeuds N_3 et N_5 implique une valeur du RSSI plus faible sur le lien $N_3 \rightarrow N_5$, que celui observé sur le lien $N_3 \rightarrow N_2$. Les valeurs retenues sont respectivement 12.5 dB et 36.1 dB. La valeur du RSSI pour le lien $N_2 \rightarrow N_5$ n'est pas représenté ici mais la valeur observée est de 33.4 dB. Nous aurions pu utiliser cette métrique pour prendre des décisions de routage. Toutefois, cette métrique ne tient pas compte de la contention au niveau du MAC. Du coup, cette métrique nous ne parait pas suffisament intéressante si on se place dans des conditions réelles.

Les figures 4(c) et 4(f) indiquent les valeurs ETX observées pour les liens $N_3 \rightarrow N_2$ et $N_3 \rightarrow N_5$. La valeur d'ETX sur le lien $N_3 \rightarrow N_5$ est supérieure à celle du lien $N_3 \rightarrow N_2$. Ceci indique la mauvaise qualité du lien concerné. Durant la phase (3), le trafic UDP ne peut être transféré convenablement, le noeud N_3 voit son buffer saturé ce qui a pour conséquence une dégradation de l'ETX sur le lien $N_3 \rightarrow N_2$. Cette dégradation n'est pas seulement due à ce phénomène mais aussi aux collisions au niveau du lien radio.

Les expérimentations confirment qu'ETX peut être une métrique intéressante pour prendre des décisions de routage. Elle est aujourd'hui disponible sur XIAN grâce



(a) RSSI $N_3 \rightarrow N_2$ (b) Throughput $N_3 \rightarrow N_2$

(c) ETX $N_3 \rightarrow N_2$ (d) RSSI $N_3 \rightarrow N_5$

(e) Throughput $N_3 \rightarrow N_5$ (f) ETX $N_3 \rightarrow N_5$

Figure 4. *Mesures réalisées avec XIAN.*

à l'extension récemment apportée : le nano-protocol. Cependant, il est très probable que la métrique ETX ne suffise pas, c'est pourquoi nous avons implémenté XIAN en fournissant un moyen simple d'ajouter de nouvelles métriques.

6. Conclusions et travaux futurs

Dans ce papier, nous présentons l'extension de XIAN, qui permet l'intégration du concept cross-layer pour des expérimentations en Wi-Fi. L'extension proposée dans ce papier permet la mise en place rapide de nouvelles métriques ayant besoin d'informations locales ou provenant du voisinage grâce au *XIAN nano-protocol*.

Afin de valider d'une part le bon fonctionnement de l'extension de XIAN, mais aussi son utilité, nous avons implémenté la métrique ETX (Expected Transmission Count). Le code complet de XIAN et de son extension est disponible en ligne [XIA].

Nous envisageons de continuer le développement de XIAN. Cette extension est très encourageante et peut être complétée par une interface permettant l'ajout de mé-

triques et de les intégrer au niveau MAC. De plus, l'introduction d'algorithmes de routage utilisant ces métriques pourrait être développée et étudiée. Cela pourrait apporter des solutions acceptables pour des routages avec qualité de service et des routages réactifs aux changements de topologies caractéristiques des réseaux ad hoc.

7. Bibliographie

[COR 99] CORSON, S. RFC 2501 : Mobile ad hoc networking (MANET) : Routing protocol performance issues and evaluation considerations. IETF (January 1999)

[CON 04] CONTI, M., MASELLI, G., TURI, G., GIORDANO, S. Cross-layering in mobile ad hoc network design. IEEE Computer (february 2004) 48–51

[KAW 03] KAWADIA, V., KUMAR, P.R. A cautionary perspective on cross layer design. IEEE Wireless Communication Magazine (july 2003)

[AIA 06] AIACHE, H., CONAN, V., LEGUAY, J., LEVY, M. Xian : Cross-layer interface for wireless ad hoc networks. In : Proc. Med-Hoc-Net. (2006)

[MAD] MADWIFI *http ://www.madwifi.org*

[DEC 03] DE COUTO, D.S.J., AGUAYO, D., BICKET, J., MORRIS, R. A high-throughput path metric for multi-hop wireless routing. In : Proc. MobiCom. (2003)

[XIA] XIAN *http ://sourceforge.net/projects/xian/*

[DEC 02] DE COUTO, D.S.J., AGUAYO, D., CHAMBERS, B.A., MORRIS, R. Performance of multihop wireless networks : Shortest path is not enough.In : Proc. HotNets, ACM SIGCOMM (2002)

[AWE 04] AWERBUCH, B., HOLMER, D., RUBENS, H. High throughput route selection in multi-rate ad hoc wireless networks. In : Proc. WONS. (2004)

[DEZ 05] M. DÉZIEL, L.L. Implementation of an IEEE 802.11 link available bandwidth algorithm to allow cross-layering. In : Proc. WiMob. (2005)

[IAN 04] IANNONE, L., KHALILI, R., SALAMATIAN, K., FDIDA, S. Cross-layer routing in wireless mesh networks. In : Proc. ISWCS. (2004)

[IPE] IPERF. *http ://dast.nlanr.net/Projects/Iperf/*

[STA] STAF Software Testing Automation Framework. *http ://staf.sourceforge.net*

Réseaux de capteurs

SensorScope, un système clef en main de surveillance de l'environnement

François Ingelrest — Guillermo Barrenetxea — Martin Vetterli

Laboratoire de Communications Audiovisuelles (LCAV)
École Polytechnique Fédérale de Lausanne (EPFL)
Station 14, EPFL, 1015 Lausanne, Suisse

{Francois.Ingelrest, Guillermo.Barrenetxea, Martin.Vetterli}@epfl.ch

RÉSUMÉ. *Étant donné l'importance des changements climatiques et leurs conséquences potentiellement dramatiques pour l'Homme, la mise au point de modèles d'évolution précis et fiables est d'une importance capitale. Malheureusement, les méthodes actuelles de collecte de données sont plutôt limitées car elles se basent sur des stations peu maniables et très chères, conduisant à un cruel manque d'observations spatiales et temporelles suffisamment denses. Les réseaux de capteurs représentent une alternative parfaitement adaptée pour répondre à ce problème, et peuvent donc avoir un impact considérable dans ce domaine. Nous présentons ici SensorScope, un projet multidisciplinaire dont le but est de fournir « clef en main » un tel système de surveillance, depuis les stations elles-mêmes jusqu'à l'infrastructure de gestion des données. Nous nous concentrons sur la collecte de ces dernières, et plus particulièrement sur les protocoles de communication qui la rendent possible. Nous détaillons également l'un de nos déploiements, effectué durant deux mois dans un glacier rocheux en Suisse, qui a permis la modélisation d'un micro-climat jouant un rôle important dans de dangereuses coulées de boue.*

ABSTRACT. *Given the importance of climate changes and their potentially dramatic consequences for human beings, designing precise and reliable evolution models is of prime importance. Unfortunately, current data collection techniques are rather limited as they make use of huge and very expensive sensing stations, leading to a lack of dense spatial and temporal observations. Wireless sensor networks are an alternative solution well-fitted to this problem, and may as such have a considerable impact on this domain. In this paper, we present SensorScope, a multidisciplinary project that aims at providing such an "out-of-the-box" monitoring system, from the sensing stations to the data management infrastructure. Here we especially focus on data gathering and on the communication protocols used for this task. We also describe one of our deployments, carried out during two months on a rock glacier in Switzerland, which led to the modeling of a micro climate that plays an important role in dangerous mud streams.*

MOTS-CLÉS : *Réseaux de capteurs, surveillance de l'environnement, déploiements.*

KEYWORDS: *Sensor networks, environmental monitoring, deployments.*

1. Introduction

Les réseaux de capteurs sont des réseaux sans fil, multi-sauts et auto-organisés, composés de *motes* déployés sur une zone à surveiller. Ces motes sont de petits objets, équipés d'un processeur et d'une radio, capables de relever diverses informations sur leur environnement par le biais de capteurs intégrés ou externes. Leur consommation énergétique est souvent une préoccupation majeure, étant donné le caractère éloigné et/ou inaccessible des zones surveillées qui rend difficile le remplacement des batteries. Les contraintes qui en résultent expliquent la présence de stations plus puissantes (des *puits*) au sein du réseau, disposant d'un moyen de communication longue portée, comme un GPRS, permettant l'envoi des données recueillies à un serveur distant.

Bien que ces réseaux possèdent de nombreuses applications, la surveillance de l'environnement est l'un des domaines où leur apport peut avoir un impact très important. Les dramatiques accidents climatiques que nous avons récemment connus ont en effet montré à quel point la connaissance de notre environnement et de son évolution peut être cruciale pour l'Homme. Toutefois, la capacité des chercheurs à améliorer celle-ci est restreinte par les méthodes actuelles de collecte des données, basées sur des stations très chères et disposant d'une capacité de stockage limitée.

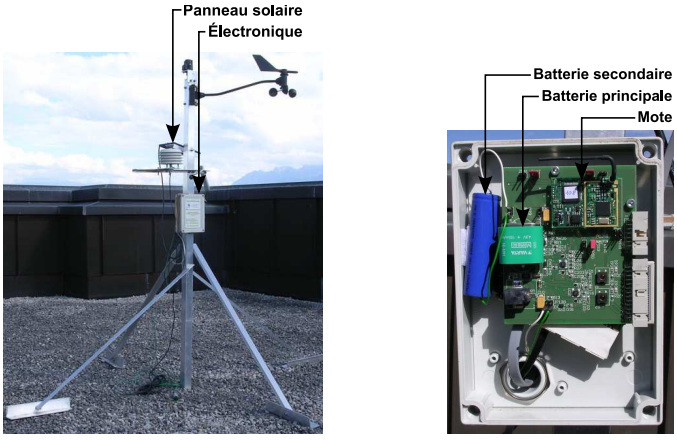
Dans cet article, nous présentons SensorScope, un système de surveillance de l'environnement. Parce qu'il repose sur un réseau de capteurs, il lève les limitations actuelles, notamment grâce à son aptitude à pouvoir transmettre immédiatement les données recueillies vers un serveur capable de les stocker durablement et en grande quantité. Celles-ci sont alors rendues publiquement disponibles en temps réel sur notre site Internet. En rassemblant les différents résultats de recherche sur les réseaux de capteurs, nous avons mis au point une pile de communication complète basée sur TinyOS, librement téléchargeable et utilisable sous une licence open-source[1]. Sa fiabilité et sa robustesse ont été largement éprouvées durant l'automne 2007, au cours duquel nous avons déployé plusieurs réseaux dans différents lieux en Suisse, dont l'un situé à 2 500 m d'altitude dans la montagne du Génépi durant deux mois.

Dans la section suivante, nous survolons les architectures matérielle et réseau de SensorScope, puis nous détaillons en section 3 les caractéristiques majeures de notre pile de communication. Dans la section 4, nous présentons les résultats obtenus durant le déploiement du Génépi cité plus tôt, puis nous concluons finalement en section 5.

2. Architecture du réseau de capteurs

Comme cela a été mentionné précédemment, le manque de relevés continus sur notre environnement bride notre capacité à le modéliser et donc à prédire son évolution. Les campagnes actuellement menées reposent sur un petit nombre de stations très chères, limitant la couverture spatiale des observations. De plus, ces stations stockent elles-mêmes les données recueillies dans une mémoire d'une taille limitée. Un réseau

1. `http://sensorscope.epfl.ch/network_code`



(a) Aperçu de la station. (b) Électronique.

Figure 1. *Une station de collecte SensorScope.*

de capteurs représente la solution idéale à ces problèmes, et permet une surveillance aussi bien en temps réel (pollution) qu'à long terme (fonte des glaces) d'événements naturels au sein de zones de taille plus ou moins grande.

Partant de cette idée, nous avons développé SensorScope, un système de surveillance de l'environnement basé sur un tel réseau. Notre but est qu'il puisse être livré « clef en main », c'est-à-dire nécessitant une connaissance et une maintenance minimales du réseau, de manière à ce qu'il puisse être utilisé aisément par des chercheurs en environnement sans qu'une armada d'informaticiens ne soit présente sur les lieux. Notre interaction au sein du projet a, de ce point de vue, été essentielle.

2.1. *Architecture matérielle*

Le mote que nous avons sélectionné est un TinyNode[2], composé d'un processeur 16 bits MSP430 cadencé à 8 MHz et d'une radio Semtech XE1205 opérant dans la bande des 868 MHz à un débit de 76 Kbps. Il dispose de 48 Ko de ROM et de 10 Ko de RAM. Nous l'avons sélectionné car il possède une grande portée, de l'ordre de 200 m en extérieur, pour une consommation électrique assez basse [DUB 06].

Quant à la station elle-même, représentée en figure 1a, elle repose sur un squelette en aluminium sur lequel sont fixés un panneau solaire et les capteurs. Sa taille (1,5 m) et ses quatre pieds la rendent stable et lui permettent de supporter les chutes de neige hivernales. La figure 1b montre que les parties électroniques sont enfermées dans une boîte hermétique placée juste au-dessus des pieds. Le prix total d'une station, incluant le mote, le panneau solaire et les capteurs, tourne autour de 900 €.

2. `http://www.tinynode.com`

2.1.1. *Alimentation électrique*

Afin d'obtenir une autonomie suffisante lors des déploiements, chaque station est équipée d'un module de gestion d'énergie solaire formé de trois composants :

Panneau solaire D'une taille de 162×140 mm et d'une durée de vie de vingt ans, il fournit une puissance maximale de 1 W en éclairage direct.

Batterie principale D'une capacité de 150 mAh et de type NiMH, elle est la source d'alimentation principale et est rechargée par le panneau solaire.

Batterie secondaire Cette batterie Li-Ion de 2 200 mAh alimente la station lorsque la capacité de la première tombe en-dessous d'un certain seuil. Elle est rechargée par la batterie principale lorsque celle-ci est à nouveau pleine.

Ce système, avec les algorithmes d'économie d'énergie de la pile de communication, rend virtuellement infinie la durée de vie d'une station. Nous avons en effet pu constater durant nos déploiements que les batteries restent toujours chargées même en cas de temps nuageux, comme nous le montrons plus loin dans la section 4.

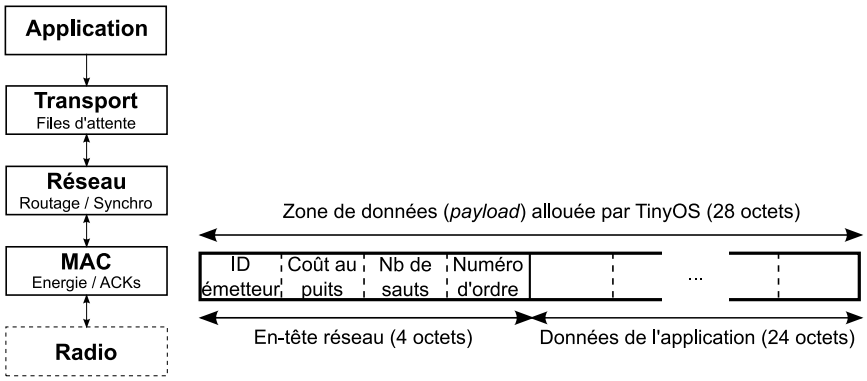
2.1.2. *Capture de données*

Jusqu'à sept capteurs peuvent être installés sur chaque station. Ceux que nous avons choisis nous permettent de mesurer neuf quantités : température de l'air et du sol, humidité de l'air, taux d'absorption et humidité du sol, vitesse et direction du vent, radiation solaire et précipitations. Afin de nous assurer de la qualité des mesures, les capteurs sont calibrés durant plusieurs jours avant d'être déployés en comparant leurs mesures à des stations de référence. Nous exigeons un coefficient de corrélation d'au moins 0.98 pour valider un capteur.

2.2. *Architecture réseau*

Pour prendre en charge le transfert des données depuis les stations jusqu'au puits, nous avons choisi de concevoir et de développer notre propre pile de communication. Nous avons fait ce choix car il est parfois très difficile de bien comprendre les interactions entre des composants existants écrits séparément, de façon à les assembler de manière correcte. De plus, ces efforts n'en valent parfois pas la peine, par exemple lorsque ces composants sont très complexes du fait de fonctionnalités qui ne sont pas nécessaires pour l'application visée.

Notre pile de communication, illustrée par la figure 2a, est implémentée sous TinyOS et suit la structure du modèle OSI. Elle n'a besoin que de quatre octets pour l'en-tête réseau, stockée dans la zone de données des paquets, ce qui laisse 24 octets pour l'application sur les 28 disponibles comme le montre la figure 2b. Nous avons choisi cette méthode, plutôt que de modifier l'en-tête standard utilisée par TinyOS, afin d'être indépendants de la radio et de ses pilotes. Il y a quatre couches différentes :



(a) La pile. (b) Format d'un paquet.

Figure 2. *La pile de communication développée pour SensorScope.*

Application Cette couche est responsable de la collecte des données (dont le niveau d'énergie des batteries) à envoyer au puits.

Transport Elle crée et reçoit les paquets réseau, puis les stocke si besoin dans une file d'attente selon leur type : les paquets de données (comme les mesures) sont routés vers le puits, ceux de contrôle (comme les ACKs) sont envoyés à un voisin en particulier. Le trafic étant faible, il n'y a pas besoin de mécanisme de contrôle de congestion. Deux champs de l'en-tête y sont également remplis : le nombre de sauts effectués par le paquet et le numéro d'ordre.

Réseau Elle passe les paquets à la couche MAC après avoir choisi un prochain saut pour ceux de données. Elle remplit les champs contenant l'identifiant de l'expéditeur et le coût de la route jusqu'au puits. Toutes les décisions de routage sont prises à ce niveau, et implémenter un nouveau protocole ne nécessite que l'écriture d'une nouvelle couche réseau. Le protocole utilisé dans SensorScope ainsi que la valeur des champs sont détaillés dans la prochaine section.

MAC Afin de réduire la consommation énergétique, elle éteint la radio dès que possible. Les paquets ne sont bien sûr envoyés que lorsque cette dernière est allumée, le reste du temps ils sont conservés et transmis plus tard. Seuls les paquets de données sont acquittés par un ACK, renvoyé au précédent saut. Nos pilotes radio ne possédant pas de mécanisme de détection d'activité du canal radio, un simple mécanisme d'attente aléatoire est utilisé pour minimiser les collisions.

3. Principales fonctionnalités de la pile de communication

Dans ce qui suit, la distance est toujours exprimée en nombre de sauts et ne désigne pas la distance euclidienne. De même, le terme *diffusion* désigne un paquet envoyé à tous les voisins directs de l'expéditeur et non pas à toutes les stations du réseau.

3.1. *Détection du voisinage réseau*

Chaque mote maintient une table de voisinage dans laquelle il stocke les voisins dont il reçoit les paquets. Nous avons choisi une méthode par *overhearing* dans l'esprit de MintRoute [WOO 03], c'est-à-dire qu'il n'y a pas de paquets dédiés mais que les motes découvrent leur voisinage grâce au trafic de données, et seul le puits doit initier le processus avec de véritables messages de signalisation. Chaque entrée de la table est associée à un coût, ce dernier étant actuellement la distance vers le puits, et une date de mise à jour. Les voisins n'étant plus assez à jour sont automatiquement supprimés.

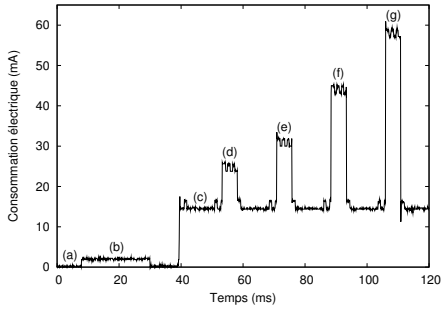
Pour éviter d'utiliser des voisins trop éloignés, une qualité de lien (QoS) est calculée pour chacun en comptant les « trous » dans les numéros d'ordre des paquets : plus les pertes sont nombreuses et plus la QoS chute. Une alternative que nous avons considérée est l'utilisation de la puissance du signal reçu, mais cette dernière varie énormément à chaque réception, et ne donne pas de bons résultats. Une remarque importante est que seuls les paquets de données sont utilisés pour le calcul de la QoS. Ainsi, puisque ces paquets sont renvoyés avec le même numéro d'ordre lorsqu'ils ne sont pas acquittés, ils font baisser la QoS des voisins qui n'arrivent pas à faire suivre les données vers le puits. La qualité reflète donc au final non seulement la capacité du voisin à être entendu, mais aussi sa capacité à router efficacement les paquets.

3.2. *Synchronisation sur une heure de référence*

Pour diverses raisons, comme une perte temporaire du signal radio, les paquets peuvent rester bloqués quelques temps à une station, ces délais empêchant le serveur de dater lui-même les données. Il est donc nécessaire que les stations possèdent une heure de référence, que nous appelons *heure réseau*, utilisée pour dater les données recueillies. Le cristal des motes étant sujet à une dérive temporelle, il est indispensable que cette heure soit régulièrement mise à jour : la dérive d'un TinyNode est par exemple de ± 20 ppm, soit une seconde toutes les 14 heures.

Dans SensorScope, des messages `SYNC_REQUEST`/`SYNC_REPLY` sont utilisés, le but étant de propager l'heure locale du puits, arbitrairement désignée comme l'heure réseau. Quand une station veut mettre à jour son horloge, elle choisit un voisin *plus proche qu'elle* du puits et lui envoie une requête. Ce dernier diffuse alors sa réponse à ses voisins qui mettent tous à jour leur heure s'ils sont *plus éloignés* du puits que lui. En procédant ainsi, l'heure se propage toujours depuis le puits qui reste la référence pour tout le réseau, et ce même si sa propre horloge dérive. Le puits envoie également régulièrement son heure au serveur qui, connaissant l'heure absolue, peut calculer le décalage et ajuster la date des données reçues afin de leur affecter une heure absolue.

Les requêtes de synchronisation sont gérées par la couche réseau, mais c'est la couche MAC qui met à jour l'heure contenue dans les paquets, éliminant ainsi les erreurs de délais [GAN 03]. Deux modes sont utilisés : un mode haute fréquence, employé lorsque l'heure réseau n'est pas connue, et un mode basse fréquence util-



Zone	Action	Valeur
(a)	Processeur inactif	0 mA
(b)	Processeur actif	2 mA
(c)	Radio allumée	15 mA
(d)	Transmission (0 dB)	25 mA
(e)	Transmission (5 dB)	31 mA
(f)	Transmission (10 dB)	44 mA
(g)	Transmission (15 dB)	58 mA

Figure 3. *Consommation électrique d'un TinyNode selon son activité.*

isé le reste du temps. D'après notre expérience, une mise à jour toutes les heures pour ce dernier mode fournit une précision d'environ 70 millisecondes, suffisante pour nos applications. Il existe des protocoles fournissant une meilleure précision, mais apportant une complexité non négligeable qui peut s'avérer inutile. Par exemple, FTSP [MAR 04] tente de compenser la dérive temporelle par des calculs de régression linéaire, calculs qui sont fortement liés au mote choisi et à sa dérive, alors que celle-ci peut considérablement varier avec les conditions climatiques telle que la température.

3.3. *Gestion de la radio*

Une bonne gestion de l'énergie est essentielle pour des déploiements de longue durée, et bien que notre système à base d'énergie solaire soit efficace, la consommation de la radio du mote est suffisamment élevée pour être prise en considération, comme le montre la figure 3. Même à l'état de « veille », c'est-à-dire uniquement à l'écoute du canal, la radio une fois allumée multiplie par sept la consommation électrique, et peut par conséquent rapidement mener à une balance énergétique négative.

Il est donc indispensable que les motes s'organisent en cycles de communication à deux états : un état *actif*, dans lequel la radio est allumée et où les paquets sont envoyés/reçus, et un état de *veille* où la radio est éteinte. L'obtention de bonnes économies d'énergie nécessite évidemment que ce dernier état soit le plus long possible. Cette organisation peut s'effectuer de deux manières :

Asynchrone Cette solution nécessite qu'un *préambule* (une suite de bits spécifique) suffisamment long soit envoyé avant chaque paquet, de manière à ce que les nœuds proches le détectent et restent éveillés en attente du paquet. B-MAC utilise ce procédé [POL 04].

Synchrone À l'opposé, cette méthode nécessite que tous les nœuds allument leur radio en même temps, ce qui permet de se passer d'un préambule. TASK utilise un tel mécanisme [BUO 05].

Nous avons choisi la seconde solution, car la première augmente de manière sensible la durée des transmissions, le préambule devant être plus long que l'état de veille, et lorsque le trafic n'est pas assez faible cela peut entraîner des congestions. Il a également été démontré [BUO 05] que la première solution entraîne une consommation énergétique un peu plus élevée et réduit donc la durée de vie totale.

Réveiller les stations simultanément est assez simple dans SensorScope, grâce au mécanisme de synchronisation présenté précédemment. Les nœuds ne connaissant pas encore l'heure réseau gardent simplement leur radio allumée, en attendant d'être synchronisés. Afin de prendre en compte une légère dérive temporelle, les stations attendent quelques dizaines de millisecondes au début de leur état actif sans rien envoyer, pour s'assurer que leurs voisins sont bien eux-mêmes éveillés. TinyOS fonctionnant par événements, cette division en cycles de communication est transparente pour les couches supérieures : si un paquet doit être envoyé alors que la radio est éteinte, il est conservé en attendant le prochain cycle et la couche du dessus attend simplement le signal `sendDone`, la durée de cette attente n'ayant aucun impact.

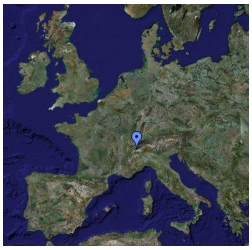
3.4. *Routage des données vers le puits*

Pour router les données, une solution généralement considérée est de créer puis de maintenir une structure en forme d'arbre dont la racine est le puits. Nous avons choisi de ne pas utiliser cette méthode, car elle implique un coût de maintenance, pour détecter les déconnections, mais également pour parvenir à un bon équilibre de la charge de routage. On peut en effet aisément imaginer qu'une station devienne le prochain saut de plusieurs autres, entraînant sa surcharge non seulement à cause de ces stations, mais aussi à cause de celles se situant plus bas dans l'arbre. Une telle situation peut par exemple se produire lorsque les nœuds sont reliés à leur meilleur parent, quelle que soit la métrique considérée, comme c'est le cas dans MintRoute [WOO 03].

Pour éviter ces problèmes, lorsqu'une station doit router des données, elle choisit toujours son prochain saut au hasard, rendant en quelque sorte le routage *opportuniste* avec une grande diversité de routes. Afin de s'assurer que les données finiront toujours par arriver, l'ensemble des prochains sauts est constitué des voisins plus proches du puits que la station elle-même. Puisqu'à chaque étape le choix est aléatoire, l'équilibre de la charge s'effectue automatiquement et chaque nœud est sollicité de la manière la plus équitable possible. Afin de valoriser les meilleurs voisins, nous avons défini deux seuils, bonne et mauvaise QoS : quand un paquet doit être routé, un voisin de bonne qualité est pris aléatoirement et s'il n'y en a pas, alors un voisin de mauvaise qualité est choisi. Les voisins en-dessous de la mauvaise qualité ne sont pas pris en compte.

4. Résultats de déploiements

Au cours de l'année 2007, nous avons effectué six déploiements en extérieur, depuis le campus de l'EPFL jusqu'à des zones de haute montagne. Nous détaillons



(a) Vue globale.

(b) Vue locale.

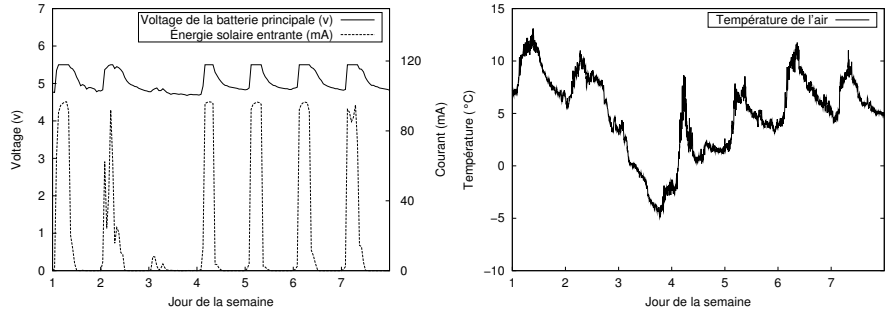
Figure 4. *Carte du déploiement du Génépi et position des stations.*

Couche	Paramètre	Valeur
Application	Échantillonnage	120 s
Réseau	Timeout d'un voisin	480 s
	Voisin de bonne qualité	$\geq 90\,\%$
	Voisin de mauvaise qualité	$\geq 70\,\%$
	Haute fréquence de synchronisation	5 s
	Basse fréquence de synchronisation	1 h ($\pm$ 72 ms de dérive)
Mac	Puissance d'émission	15 dB
	Durée de l'état actif	12 s
	Durée de l'état de veille	108 s

Tableau 1. *Paramètres utilisés durant la campagne du Génépi.*

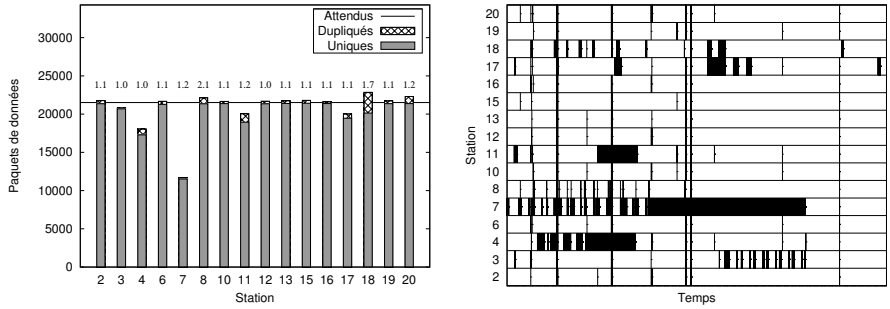
ici l'un de nos plus importants déploiements, réalisé dans un glacier rocheux situé à 2 500 m d'altitude sur la montagne du Génépi en Suisse. Ce site a été choisi car, lors de pluies intenses, il est la source de coulées de boue ayant tué plusieurs personnes ces dernières années. Les autorités cantonales en charge de la sécurité n'ayant à ce moment aucun modèle climatique du site, elles nous ont demandé d'y déployer un réseau afin de corréler les précipitations avec le vent et la température en fonction de la forme du terrain. Elle nous ont fourni toute l'aide technique pour cela, incluant un hélicoptère pour le transport du matériel et des personnes. Ce déploiement a été effectué durant les derniers jours du mois d'août 2007 et a duré plus de deux mois.

Les 16 stations ont été déployées sur une zone de 500×500 m comme le montre la figure 4. Leur position a été choisie avec attention, afin d'obtenir des observations significatives : la station 20 a ainsi été placée dans la zone de rupture du glacier et la 11 dans la pente de rochers. Afin de transmettre les données à nos serveurs, le puits, situé près de la station 3, était équipé d'un module GPRS que nous utilisions pour la première fois. Comme nous le montrons plus loin, son fonctionnement a été problématique et a causé la perte de quelques données, notamment à cause de la faible connectivité sur le site. Les différents paramètres utilisés sont fournis par le tableau 1.



(a) Énergie disponible. (b) Température de l'air observée.

Figure 5. *Une semaine dans le système d'énergie solaire d'une station.*

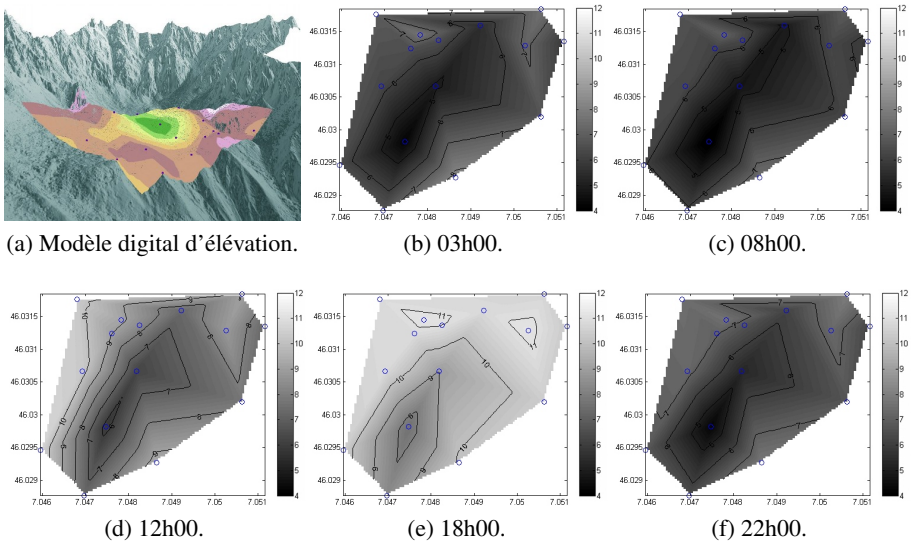


(a) Paquets de données reçus et distance moyenne des stations au puits. (b) Corrélation temporelle des pertes.

Figure 6. *Statistiques des données reçues durant un mois complet.*

Cette campagne a été une bonne occasion de tester l'autonomie des stations en conditions réelles. La figure 5 montre ainsi la variation d'énergie de la station 15 pendant une semaine complète, l'énergie solaire recueillie ainsi que la corrélation avec la température observée par cette même station. Les mesures commencent le matin du 16 septembre 2007 à 6h00, période durant laquelle le lever du soleil était autour de 6h00 et son coucher autour de 21h00, et montrent la consommation totale de la station, incluant ses capteurs. On peut nettement voir que la batterie principale se vide lentement la nuit puis se recharge complètement la journée. Au troisième jour, le temps a été très nuageux, impliquant une chute brutale de la température, mais malgré le faible ensoleillement la charge fut suffisante. Durant toute la semaine, la batterie secondaire n'a pas du tout été sollicitée, ce qui prouve l'efficacité du système.

La figure 6 donne les statistiques de la collecte des données durant un mois à partir du 10 septembre 2007, et montre que la plupart des paquets a été recueillie pour la



(a) Modèle digital d'élévation. (b) 03h00. (c) 08h00.

(d) 12h00. (e) 18h00. (f) 22h00.

Figure 7. *Modèle d'élévation (0.5×0.5 m) et distribution spatiale de la température de l'air le 2 octobre 2007.*

majeure partie des stations. Certaines d'entre elles, notamment les stations 4, 7, 11 et 17, ont été victimes de problèmes matériels, tels que des court-circuits, qui ont entraîné la perte d'un certain nombre de paquets consécutifs. Ce phénomène est très visible sur la figure 6b où l'on peut voir la corrélation temporelle des pertes par station, chaque trait représentant un paquet manquant. Cette figure met également en avant les problèmes liés au GPRS, ces derniers se manifestant par des pertes simultanées pour toutes les stations. Nous savons à présent qu'elles ont été causées par un bug dans le programme du GPRS, entraînant la suppression des paquets mémorisés lors d'une perte de la connexion au réseau cellulaire.

Finalement, la figure 7 fournit une partie des résultats de l'analyse actuellement menée par nos collaborateurs en mécanique des fluides de l'environnement. Elle montre la position des stations sur le modèle d'élévation digitale du glacier rocheux, au centre duquel on peut voir la vallée où le permafrost est le plus épais (jusqu'à 15 m) et où les coulées de boue prennent leur source. Les autres figures montrent la distribution spatiale de la température de l'air le 2 octobre 2007, journée durant laquelle le ciel était parfaitement dégagé. Chaque valeur est la moyenne d'une heure de mesures.

Ces résultats montrent que même en cas de ciel ensoleillé, la température est toujours basse dans la vallée. En effet, tandis que la variation de température est autour de 5°C à la bordure du site, elle n'est que de 2°C dans la vallée. Cette observation est intéressante car toutes les stations étaient exposées de la même manière au soleil, et auraient donc dû reporter les mêmes valeurs. Cette différence s'explique par l'influence de l'épaisse couche de glace du permafrost, conservant la température à une

valeur assez basse quelle que soit l'exposition au soleil. Ce micro-climat est actuellement considéré comme l'un des éléments clefs dans la prédiction des coulées de boue.

5. Conclusion

À travers les déploiements effectués, SensorScope est devenu un projet clef, fusionnant les technologies des réseaux de capteurs et celles de la recherche environnementale. La campagne du Génépi a été très enrichissante, non seulement en termes de données collectées mais également sur le plan de l'expérience acquise. Les défauts de jeunesse auxquels nous avons fait face, notamment liés au matériel, sont en cours de résolution et ne devraient plus poser de problème à l'avenir. L'implémentation d'un système de reprogrammation à distance tel que Deluge [HUI 04] est également à l'étude, afin de minimiser les déplacements sur le terrain une fois le réseau déployé.

Remerciements

Ces travaux ont été en partie financés par NCCR-MICS, par le projet Européen FP 6 WASP, ainsi que par Microsoft Research.

6. Bibliographie

[BUO 05] BUONADONNA P., GAY D., HELLERSTEIN J., HONG W., MADDEN S., « TASK : Sensor Network in a Box », *Proceedings of the IEEE European Workshop on Wireless Sensor Networks and Applications (EWSN)*, janvier 2005.

[DUB 06] DUBOIS-FERRIÈRE H., MEIER R., FABRE L., METRAILLER P., « TinyNode : A Comprehensive Platform for Wireless Sensor Network Applications », *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, avril 2006.

[GAN 03] GANERIWAL S., KUMAR R., SRIVASTAVA M., « Timing-Sync Protocol for Sensor Networks », *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, novembre 2003.

[HUI 04] HUI J., CULLER D., « The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at a Scale », *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, novembre 2004.

[MAR 04] MARÓTI M., KUSY B., SIMON G., LÉDECZI A., « The Flooding Time Synchronization Protocol », *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, novembre 2004.

[POL 04] POLASTRE J., HILL J., CULLER D., « Versatile Low Power Media Access for Wireless Sensor Networks », *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, novembre 2004.

[WOO 03] WOO A., TONG T., CULLER D., « Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks », *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, novembre 2003.

Etude et extension des relais de couverture de surface dans les réseaux de capteurs

Antoine Gallais — Jean Carle

LIFL CNRS UMR 8022, Univ. Lille 1, INRIA Futurs, France
{Antoine.Gallais, Jean.Carle}@lifl.fr

RÉSUMÉ. *L'objectif des protocoles de couverture de surface est de mettre en veille des capteurs redondants tout en assurant une couverture totale de la zone par les nœuds restant allumés. La connexité de l'ensemble des nœuds actifs doit également être préservée dans la mesure où il est impératif d'assurer l'acheminement des rapports de surveillance vers les stations puits. La plupart des solutions existantes ne considèrent pas ces deux problèmes comme un seul et sont rarement évaluées sous des conditions réalistes. Dans cet article, nous proposons une analyse approfondie et des améliorations d'un protocole localisé maintenant la couverture de zone par des ensembles connectés en se basant sur des relais de couverture de surface (SCR). Dans un premier temps, nous avons amélioré l'une des phases critiques de l'algorithme ; la sélection des relais. Ceci nous a permis de diminuer le nombre de nœuds actifs sans altérer la couverture fournie. Nous avons ensuite soulevé le problème de sa résistance à des conditions de communication plus réalistes. Nous verrons alors comment de nouvelles modifications durant la phase de sélection des relais nous ont permis d'obtenir un protocole robuste et facilement configurable.*

ABSTRACT. *Area coverage protocols aim at turning off redundant sensor nodes while ensuring full coverage of the area by the remaining active nodes. Connectivity of the active nodes subset must also be provided so that monitoring reports can reach the sink stations. Existing solutions hardly address these two issues as a unified one and very few are robust to non ideal physical conditions. In this paper, we propose a deep analysis and some enhancements of a localized algorithm for area coverage, based on Surface Coverage Relays (SCR) and able to build connected active nodes sets that fully cover the area. We first enhanced the critical phase of our protocol (the relay selection) and show that the number of active nodes can be drastically reduced. We then raise the issue of the robustness of the protocol once a realistic physical layer is simulated. We will show how another modification of our relay selection phase led us to an even more robust and configurable protocol.*

MOTS-CLÉS : *Réseaux de capteurs sans fils, couverture de surface, algorithme localisé configurable, couche physique réaliste*

KEYWORDS: *Wireless sensor networks, area coverage, configurable localized algorithm, realistic physical layer.*

1. Introduction

Les réseaux de capteurs sans fils sont composés de centaines d'objets équipés d'une batterie, d'une unité d'acquisition d'information et sont capables de communiquer sans fil. Déployés sur des zones distantes ou sensibles, ils deviennent alors à usage unique en raison de l'impossibilité de remplacer ou recharger leurs batteries. La consommation d'énergie est alors distribuée en exploitant la redondance induite par le déploiement aléatoire ; des nœuds sont actifs tandis que d'autres demeurent passifs, économisant ainsi leur énergie. Cette topologie dynamique ne doit pourtant pas mettre en péril l'application de surveillance. C'est pourquoi les nœuds actifs doivent couvrir une zone aussi large que l'ensemble des capteurs déployés. De plus, il est indispensable d'assurer l'acheminement des rapports en provenance des capteurs vers les stations puits chargées de stocker et de traiter les données. L'ensemble des nœuds actifs doit également être connecté. Dans [CAR 05], nous avons proposé un protocole localisé capable de maintenir une couverture de surface totale à l'aide d'ensembles connectés. Une première phase de découverte de voisinage suivie d'une autre de sélection des relais de couverture de surface sont nécessaires au processus de prise de décision exécuté par chaque nœud.

Dans cet article, nous revenons sur la phase de sélection de relais que nous avons mise en place. Cette solution nous permet de réduire davantage le nombre de nœuds actifs sans introduire de messages ou d'informations supplémentaires. La majeure partie des solutions au problème de couverture de surface considère des communications idéales (chaque nœud ayant un rayon de communication théorique au-delà duquel il ne peut rien recevoir ni transmettre directement). En réalité, il existe de nombreux aléas dans les communications sans fils qui ne sont pas considérés dans ces solutions. Nous avons soumis notre algorithme à des couches physiques non idéales (i.e. toute transmission ayant une probabilité donnée de se dérouler correctement). Nous montrerons qu'il s'en trouve très légèrement atteint, assurant une couverture moindre qu'avec une couche physique idéale. Nous avons alors introduit un nouveau paramètre, intervenant au cours de la sélection des relais, et qui nous permet de renforcer notre protocole. Nous montrerons également comment l'ajout de ce paramètre peut nous aider à configurer plus finement l'ordonnancement d'activité des capteurs déployés en fonction des besoins de l'application de surveillance.

2. Préliminaires

Nous supposons que les capteurs sont déployés aléatoirement et demeurent statiques. La plupart des solutions existantes supposent que les nœuds connaissent leur position exacte. La même hypothèse est reprise ici étant donné le problème de positionnement a déjà été largement étudié [BAC 05]. La position exacte peut être utilisée comme un identifiant unique pour chaque nœud. Nous supposons également que les capteurs sont synchronisés en temps de façon à organiser l'activité en tours successifs. Là encore, de nombreuses solutions ont été proposées dans ce domaine (voir [MAR 04] par exemple).

2.1. *Modèles de communication et de surveillance*

Un réseau est modélisé par un graphe $G = (V, E)$, V étant l'ensemble des sommets et $E \subseteq V^2$ celui des arêtes. Une arête entre deux sommets u et v existe si u est physiquement capable d'envoyer un message à v. Le voisinage de u, noté $\mathrm{N}(u)$, est défini comme suit :

$$\mathrm{N}(u) = \{v \in V \mid v \neq u \wedge (u, v) \in E\}. \quad [1]$$

Chaque objet possède un rayon de communication théorique R_c ainsi qu'un rayon de surveillance R_s. Nous noterons $\mathrm{S}(u)$ la surface couverte par un nœud u et $\mathrm{S}(A)$ celle couverte par un ensemble de sommets $A = \{a_1, a_2, \ldots, a_n\}$, tel que :

$$\mathrm{S}(A) = \bigcup_{i=1}^{i=|A|} \mathrm{S}(a_i). \quad [2]$$

2.2. *Modèles de canal radio*

Soit un graphe $G = (V, E)$ et la distance euclidienne entre les nœuds u et v $\mathrm{dist}(u, v)$, le modèle du disque unitaire (*unit disk graph*) définit E de la manière suivante :

$$E = \{(u, v) \in V^2 \mid u \neq v \wedge \mathrm{dist}(u, v) \leq R_c\}, \quad [3]$$

Cependant, dans un contexte pratique, chaque lien de communication a une probabilité d'exister, qui dépend de nombreux facteurs tels que la puissance d'émission et la distance émetteur/récepteur. Durant notre évaluation de performances, nous serons amenés à remplacer le modèle du disque unitaire par celui dit du masquage lognormal (*lognormal shadowing*), transformant ainsi G en un graphe valué, où le poids de chaque arête $(u, v) \in E$ correspond à la probabilité de réception correcte $\mathrm{p}(\mathrm{dist}(u, v))$ pour deux nœuds u et v. Une fonction d'approximation $\mathrm{P}(x)$ est décrite par Kuruvila et al. [KUR 04] :

$$\mathrm{P}(x) = \begin{cases} 1 - \frac{(\frac{x}{R_c})^{2\alpha}}{2} & \text{si } 0 < x \leq R_c, \\ \frac{(\frac{2R_c - x}{R_c})^{2\alpha}}{2} & \text{si } R_c < x \leq 2 \times R_c, \\ 0 & \text{sinon,} \end{cases} \quad [4]$$

α étant fonction de l'environnement et x étant la distance considérée.

Cette fonction suppose que la probabilité de réception correcte à distance R_c est toujours égale à P(R_c) = 0.5 (voir Fig. 1).

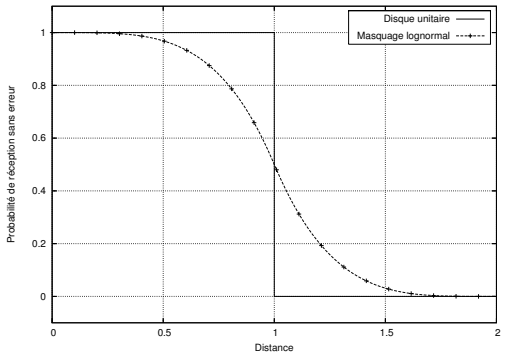


Figure 1. *Les deux modèles de couche physique considérés ($R_c = 1, \alpha = 2$).*

Ce modèle, bien que plus réaliste que celui du disque unitaire, ne considère pas tous les facteurs caractérisant une véritable liaison sans fil. Si certains aspects, tels que la modulation ou les effets d'évanouissement, ne sont pas pris en compte directement, l'approximation employée se rapproche néanmoins de résultats précédemment obtenus dans des conditions réelles [GAL 06].

3. Travaux existants

Tian et Georganas [TIA 05] ont proposé un algorithme d'ordonnancement d'activité permettant à des capteurs synchronisés et aléatoirement déployés de régulièrement recenser leurs voisins puis d'attendre un temps donné à l'écoute d'éventuels messages de retraits. Ces derniers sont envoyés par un nœud ayant atteint la fin de son temps d'attente et qui est complètement couvert. La connexité est assurée du fait d'un rapport fixe entre les rayons de surveillance et de communication ($R_s \leq R_c$). Il est prouvé que cette hypothèse combinée avec une couverture de zone totale induit nécessairement un ensemble connecté [XIN 05].

Récemment, un algorithme aléatoire a été proposé dans [YEN 07]. Reposant sur un modèle de chaînes de Markov, cette solution permet à chaque capteur de devenir passif avec une probabilité donnée. Leur analyse montre qu'il est possible de configurer cette dernière pour assurer une couverture de surface totale ainsi que la connexité de l'ensemble, ce quel que soit le ratio entre R_c et R_s. Néanmoins, tout changement de configuration nécessite une nouvelle analyse, ce qui pourrait se révéler un obstacle à une utilisation au sein de réseau dynamiques.

4. Le protocole SCR-CADS

Alors que le maintien de la couverture de zone pourrait ne pas être une contrainte stricte, celui de la connexité s'avère souvent plus crucial en raison des informations qu'il faut acheminer. C'est pour cette raison que nous sommes revenus sur notre protocole SCR-CADS, prouvé comme assurant une couverture totale par des ensembles connectés. Cette solution consiste en trois étapes successives effectuées par chaque nœud (découverte de voisinage, sélection des relais et prise de décision) que nous rappelons brièvement.

4.0.1. *Découverte de voisinage*

Cette phase est classique ; des messages *hello* sont échangés, contenant la position du nœud émetteur.

4.0.2. *Sélection des relais de couverture de surface*

En étudiant la couverture de point, Jacquet et al. [ADJ 05] ont utilisé les relais multipoints (MPR) [QAY 02]. Chaque nœud sélectionne un sous-ensemble de son voisinage (l'ensemble MPR), suffisant à communiquer avec l'ensemble des voisins à deux sauts. Chaque nœud u doit ensuite envoyer son ensemble $MPR(u)$ à ses voisins afin de permettre la prise de décision. Cette dernière s'effectue selon une règle simple, permettant de préserver la connexité de l'ensemble des nœuds actifs.

Comme notre but était d'assurer la connexité et la couverture, notre solution s'en inspire directement en proposant une définition des relais adaptée à la couverture de zone ; chaque nœud u calcule un ensemble de relais de couverture de surface $SCR(u)$ devant couvrir une zone identique à celle couverte par l'ensemble des voisins :

$$\forall u, \exists\, \mathrm{SCR} \mid S(\mathrm{SCR}(u)) \backslash S(u) \;=\; S(N(u)) \backslash S(u). \qquad [5]$$

Chaque nœud doit être capable de calculer son propre ensemble SCR. Notons que construire un ensemble de taille minimale est équivalent au problème du « weighted set cover » qui est NP-complet [KAR 72].

Initialement, nous avions proposé une heuristique reposant sur un calcul d'isobarycentres. Le voisin le plus lointain est d'abord considéré et nécessairement ajouté à l'ensemble SCR vide. Puis, afin d'éviter un calcul coûteux (quel est le voisin qui apporte le plus de couverture supplémentaire par rapport à celle déjà fournie par l'ensemble SCR ?), nous avions proposé de choisir le voisin le plus éloigné de l'isobarycentre des nœuds constituant l'ensemble des relais.

Nous proposons ici une nouvelle méthode qui consiste à trier les voisins par ordre décroissant de distance et à les considérer dans cet ordre lors du calcul de SCR. Si un nœud couvre une portion de zone encore non couverte par l'ensemble SCR, alors le voisin doit appartenir à cet ensemble. Une fois les ensembles SCR calculés et envoyés aux voisins, chaque capteur peut appliquer la règle de décision.

4.0.3. *Prise de la décision d'activité*

Dans le protocole MPR-DS [ADJ 05], les nœuds appliquent une règle simple basée sur un identifiant (ou une clef) supposé unique pour chaque nœud.

Definition 1 *Tout nœud u dont la clef est la plus petite du voisinage ou qui appartient à l'ensemble des relais du voisin ayant la plus petite clef doit être actif.*

Comme déjà montré dans [ADJ 05], cette règle suffit à garantir la connexité de l'ensemble obtenu. La même conclusion a été démontrée lorsque qu'elle est appliquée avec des ensembles SCR et mène également à une couverture de zone totale [CAR 05].

Maintenant que nous avons introduit SCR et présenté une nouvelle heuristique de sélection des relais, nous allons évaluer les performances de notre solution avec différents modèles de communication. Comme nous l'avons déjà évoqué, il est important de considérer des hypothèses réalistes afin de tester la validité de notre algorithme.

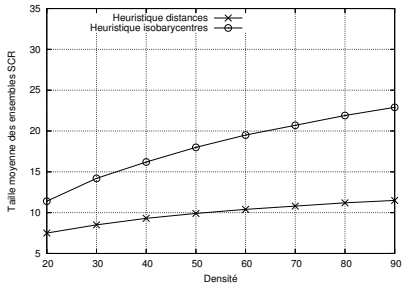
5. Evaluation de performances

Les résultats expérimentaux ont été obtenus à partir de graphes générés aléatoirement au sein d'un simulateur à événements discrets. Les nœuds sont déployés sur une zone carrée de 50×50. Les rayons R_c et R_s sont fixés à 10. Dans nos simulations, la densité du réseau se réfère au nombre moyen de nœuds par zone de communication. Chaque itération débute par la phase de découverte du voisinage, suivie de la sélection des relais et enfin de la prise de décision. La période suivante concerne les nœuds actifs uniquement (surveillance et trafic des informations collectées) et n'est pas reprise ici. Pour chaque densité, un nombre d'itérations suffisant est simulé. Par suffisant, nous entendons que 95% des résultats se trouvent dans un intervalle de confiance suffisamment étroit. Enfin, lorsque les nœuds doivent évaluer la zone couverte par un ensemble de capteurs, ils utilisent un outil d'évaluation de couverture, basé sur les intersections des disques de surveillance [ZHA 05, XIN 05].

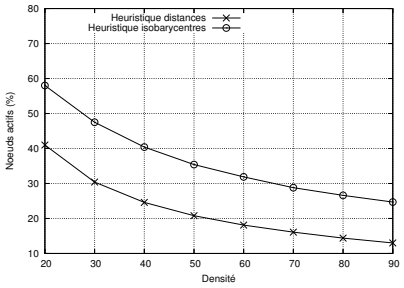
5.1. *Sous des conditions de communication idéales*

Nous donnons pour commencer des résultats complémentaires au sujet du protocole SCR lorsqu'il est simulé avec le modèle du disque unitaire. Nous parlons alors de conditions idéales dans la mesure où aucun message ne peut être perdu, que ce soit en raison de collisions ou de l'environnement de transmission.

Comme nous l'avons déjà mentionné, l'algorithme a été prouvé comme maintenant à la fois la couverture de surface et la connexité de l'ensemble des nœuds actifs. Nous nous concentrons donc à présent sur la taille des ensembles de relais suite à l'introduction de notre nouvelle heuristique.



(a) Réduire la taille des ensembles de relais.



(b) Diminuer la proportion de nœuds actifs.

Figure 2. *Principaux avantages de l'heuristique des distances pour la sélection des relais.*

5.1.1. *Taille des ensembles SCR*

Nous avons calculé la taille moyenne des ensembles SCR afin de comparer les deux heuristiques de sélection proposées. Cette donnée est importante d'abord dans la mesure où le nombre de nœuds actifs en dépend nécessairement et ensuite parce qu'elle détermine la taille des données de contrôle à envoyer dans les messages à l'issue de la sélection. Concernant le premier point, tout nœud ayant la plus petite clef de son voisinage est inéluctablement actif. Ce nombre varie donc en moyenne uniquement avec la densité. En revanche, plus les ensembles SCR sont grands plus la probabilité d'appartenir à celui du voisin ayant la plus petite clef l'est également, rendant par conséquent plus probable le fait de devoir être actif. Pour ce qui est du second point, étant donné que les ensembles SCR doivent être communiqués aux voisins à un saut, moins ils seront imposants, moins les messages seront grands, et la consommation énergétique du capteur liée au protocole sera réduite.

Nous avons pu observer que la nouvelle heuristique basée sur le calcul de distances procurait de meilleurs résultats que celle reposant sur les isobarycentres. La figure 2(a) nous montre une taille moyenne des ensembles générés avec la nouvelle méthode allant de 8 à 11 voisins, pour les densités 20 et 90 respectivement. Rappelons que la solution utilisant les barycentres conduisait à 11 voisins contenus dans l'ensemble SCR à densité 20 et jusque 24 à densité 90, soit plus du double de ce que nous obtenons aujourd'hui.

Voyons à présent si cette forte réduction de la taille des ensembles de relais a un impact similaire sur les proportions de nœuds actifs.

5.1.2. *Nœuds actifs*

La figure 2(b) nous montre les pourcentages de nœuds actifs en fonction de la densité du réseau. Ces proportions étaient par exemple de 24.7% à densité 90 avec l'heuristique des barycentres et ne sont plus maintenant que de 13% pour la même densité, avec notre nouvelle méthode. Ces résultats confirment l'importance de la sélection des relais et de la méthode qui y est employée ; la taille des données utiles dans les messages de contrôle peut être divisée par deux et le nombre de nœuds pouvant demeurer passifs à chaque tour est augmenté de 15%. Nous pensons donc que notre protocole pourrait avoir des effets bénéfiques en terme de consommation énergétique s'il venait à être employé dans la réalité. Avant cela, nous avons évalué sa capacité à résister aux aléas du canal radio.

5.2. *Avec une couche physique réaliste*

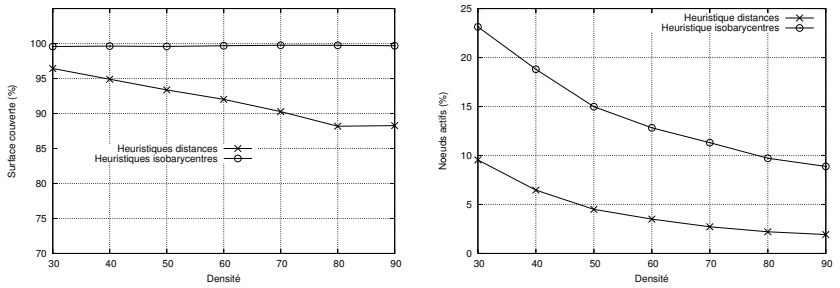
Nous évaluons maintenant notre protocole SCR-CADS avec une couche physique réaliste. Nous voulons évaluer sa robustesse et voir si des ajustements sont nécessaires. A présent, les liens de communication sont probabilistes ; deux nœuds communiquent avec une probabilité donnée fournie par la fonction 4. Ni retransmissions, ni acquittements n'ont été introduits dans cette étude sur des réseaux très denses. Nous nous sommes donc placés dans le pire cas où chaque message n'est envoyé qu'une seule fois.

Dans notre algorithme, une fois la découverte du voisinage effectuée et les ensembles SCR calculés, ils sont envoyés aux voisins à un saut afin de permettre la prise de décision basée sur la règle 1. Tant que le modèle du disque unitaire est employé, tous ces messages envoyés sont reçus par les voisins à un saut et des ensembles couvrants et connectés sont calculés. Observons à présent l'impact d'une couche physique réaliste sur la couverture de surface.

5.2.1. *Couverture de surface*

La figure 3(a) donne la surface couverte par les nœuds actifs en fonction de la densité du réseau. Si la couverture de zone était totale avec des communications parfaites, ce n'est plus le cas lorsque des aléas sont introduits dans les transmissions. Le pourcentage de couverture passe de 95% à densité 30 à moins de 90% à densité 80. Ces pertes ne sont pas aussi importantes que celles que nous avons déjà pu remarquer sur d'autres protocoles de couverture de surface [GAL 07]. De plus, une couverture garantie à 90% pourrait suffire dans de nombreuses applications d'observation ou de surveillance.

Expliquons maintenant pourquoi cet impact est si faible. Durant la phase de découverte de voisinage, les messages *hello* d'une partie des voisins pourrait ne pas être reçus par un nœud donné u. Cela ne fait pourtant qu'augmenter la probabilité qu'a u de penser que sa clef est la plus petite du voisinage. Ainsi, moins u découvre de voisins, plus ses chances d'être actif sont grandes. Imaginons alors qu'aucun message



(a) Perte de couverture.

(b) Réduction des nœuds actifs (%).

Figure 3. *L'algorithme SCR-CADS avec une couche physique réaliste (LNS, voir formule 4, avec $\alpha = 2$).*

ne puisse être transmis, alors tous les nœuds seront actifs et la surface entièrement couverte. Ce n'est pas le cas ici et les pertes de couverture ne peuvent pas s'expliquer par la phase de découverte de voisinage.

Voyons donc la phase de sélection des relais. Une fois calculés, les relais sont envoyés aux voisins à un saut. Notons qu'à présent, la notion de voisinage à un saut est difficile à cerner et nous préférons dire que ce message est émis et potentiellement reçu par tout nœud situé dans un rayon de $2 \times R_c$. Si certains ne sont jamais reçus par un nœud u, la conséquence sera que u ne connaîtra qu'un sous-ensemble de ses véritables sélecteurs SCR. Si l'un de ceux que u ignore se trouve être celui à la clef la plus petite du voisinage, alors u pensera ne pas avoir été sélectionné par ce dernier et décidera d'être passif, mettant en péril la couverture de surface. Ceci explique les pertes de couverture observées sur la figure 3(a) lorsque l'heuristique des distances est employée. L'impact est beaucoup moins important avec la méthode basée sur le calcul de barycentres, car elle génère toujours des ensembles SCR de plus grande taille et par conséquent davantage de nœuds actifs, comme nous le verrons par la suite. Notons enfin que si un message de relai est reçu par u depuis un nœud v jusqu'à présent inconnu (pas de message *hello* de v reçu par u durant la première phase), alors u devra considérer ce message. En effet, v avait peut-être reçu le message *hello* de u et pourrait l'avoir sélectionné comme relai de couverture. Finalement, seul un message est crucial ; celui contenant l'ensemble SCR du voisin ayant la plus petite clef du voisinage. Les aléas de communication peuvent ensuite être sources de nouvelle information.

5.2.2. *Nœuds actifs*

Généralement, lorsque des pertes de couverture se produisent, elles sont causées soit par une diminution du nombre de nœuds actifs (trop peu restent pour pouvoir couvrir la zone entièrement), soit par une incohérence spatiale (assez de nœuds actifs

mais inégalement distribués). Le deuxième cas peut difficilement se produire en raison de l'homogénéité de la distribution initiale. Effectivement, la figure 3(b) nous montre que le nombre de nœuds actifs est largement diminué, passant de moins de 10% à densité 30 à seulement 2% à densité 90.

Comme nous l'avons vu précédemment, cette forte baisse des proportions de nœuds actifs ne met pas en grand péril l'application de surveillance, démontrant ainsi l'inhérente robustesse de notre solution.

5.3. *Davantage de robustesse et de configurabilité*

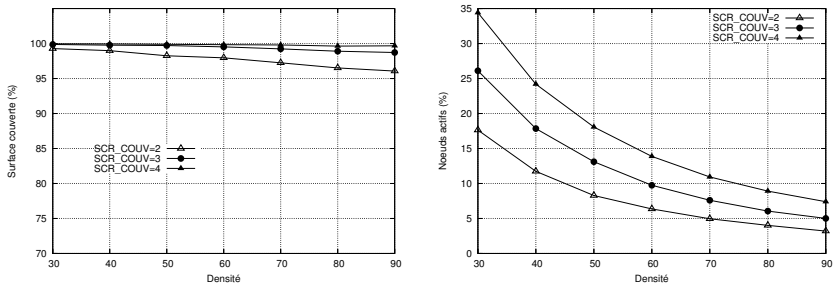
Bien que l'impact d'une couche physique réaliste soit moindre, nous avons observé une forte diminution des nœuds actifs, lui-même essentiellement influencé par la taille des ensembles de relais. Nous essayons à présent de donner une propriété de configurabilité à notre algorithme afin de pouvoir influencer facilement le nombre de nœuds actifs et par conséquent la proportion de zone couverte (pouvant ainsi corriger les pertes occasionnées par les aléas des transmissions).

Nous avons introduit un nouveau paramètre dit *de couverture* et noté SCR_COUV. Il est utilisé durant la sélection des relais en imposant le nombre de relais devant couvrir chaque point de la zone couverte par l'ensemble des voisins. Autrement dit, pour chaque voisin considéré lors de la sélection, si la zone qu'il couvre n'est pas k-couverte (avec $k = SCR_COUV$) par l'ensemble des relais déjà sélectionnés alors il doit faire partie de l'ensemble SCR. Il existe plusieurs définitions de la k-couverture de surface. Notre approche est de considérer une zone comme k-couverte sitôt que tout point appartenant à cette surface est à portée de surveillance de k capteurs distincts.

Ceci devrait logiquement nous aider à augmenter les proportions de nœuds actifs, étant donné qu'elles sont grandement déterminées par les tailles de ensembles SCR. Celles-ci sont nécessairement augmentées par notre contrainte de k-couverture de surface. Pour les résultats présentés dans cette section, l'heuristique des distances est utilisée lors de la sélection des relais.

5.3.1. *Couverture de surface*

Le but premier du paramètre SCR_COUV est d'être capable de résister davantage aux pertes de messages en augmentant intentionnellement la redondance des nœuds actifs. La figure 4(a) nous montre la couverture de surface obtenue pour plusieurs valeurs du paramètre de couverture ($SCR_COUV \in \{2, 3, 4\}$). Aussitôt que $SCR_COUV > 1$, la couverture est d'au moins 95%. Ceci montre que même en présence d'aléas durant les transmissions sans fils, notre algorithme peut fournir des résultats satisfaisants. Notre nouveau paramètre peut également nous aider à configurer la sélection des relais. Jusqu'à présent, nous ne disposions que de nos deux heuristiques. Le problème est qu'il est difficile aujourd'hui de changer dynamiquement le code déployé sur un capteur communicant. Changer la valeur du paramètre de couverture est en revanche plus aisé.



(a) Augmenter la robustesse aux aléas du canal radio.

(b) Proportions de nœuds actifs configurables avec la sélection de relais étendue.

Figure 4. *Configurabilité du protocole SCR-CADS (LNS, voir formule 4, avec $\alpha = 2$).*

5.3.2. *Nœuds actifs*

Nous nous sommes enfin demandés à quel point la valeur de SCR_COUV pouvait influencer les proportions de nœuds actifs. D'après la figure 4(b), lorsque la valeur de SCR_COUV est égale à 2 (ce qui est suffisant pour assurer plus de 95% de surface couverte), nous obtenons entre 17% et 4% de nœuds actifs. Les courbes montrent que quelle que soit la densité du réseau, modifier la valeur de ce paramètre suffit à influencer clairement la redondance de surveillance.

6. Conclusion

Nous avons amélioré notre protocole SCR-CADS afin d'obtenir une phase de sélection des relais de couverture plus adaptée aux contraintes que nous pensons retrouver dans un environnement de déploiement réel. Avoir une solution localisée simple et pouvoir régler la redondance des nœuds actifs tout en conservant une robustesse aux aléas du canal radio sont pour nous autant de raisons qui font de cet algorithme un candidat crédible pour des déploiements réels.

Sitôt après avoir introduit une couche physique plus réaliste, nous n'avons plus évoqué le problème de la connexité du graphe, principalement car il est difficile de caractériser cette dernière. Nous aimerions maintenant étudier cet aspect afin de savoir si notre algorithme s'avère toujours satisfaisant pour des déploiements réels. Dans ce cadre, l'introduction de modèles de communication plus réalistes, intégrant par exemple les interférences, fait naturellement partie de nos ambitions. De même, si les besoins en synchronisation ou en localisation venaient à s'avérer des obstacles dans nos scénarios de déploiements (en termes de passage à l'échelle ou de coûts), il nous faudrait évaluer l'impact d'erreurs dans ces calculs et éventuellement de quelle manière notre protocole pourrait s'affranchir de ces hypothèses.

7. Bibliographie

[ADJ 05] ADJIH C., JACQUET P., VIENNOT L., « Computing connected dominated sets with multipoint relays », *Ad Hoc and Sensor Wireless Networks journal (AHSWN)*, vol. 1, n° 3, 2005, p. 27–39.

[BAC 05] BACHRACH J., TAYLOR C., « Localization in sensor networks », *Handbook of Sensor Networks (I. Stojmenović, ed.), Wiley*, , 2005, p. 277–310.

[CAR 05] CARLE J., GALLAIS A., SIMPLOT-RYL D., « Preserving Area Coverage in Wireless Sensor Networks by using Surface Coverage Relay Dominating Sets », *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, Cartagena, Spain, 2005, p. 347–352.

[GAL 06] GALLAIS A., PARVERY H., CARLE J., GORCE J.-M., SIMPLOT-RYL D., « Efficiency Impairment of Wireless Sensor Networks Protocols under Realistic Physical Layer Conditions », *Proceedings of 10th IEEE International Conference on Communication Systems (ICCS)*, Singapore, 2006.

[GAL 07] GALLAIS A., INGELREST F., CARLE J., SIMPLOT-RYL D., « Preserving Area Coverage in Sensor Networks with a Realistic Physical Layer », *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, 2007, short paper.

[KAR 72] KARP R. M., « Reducibility among combinatorial problems », MILLER R. E., THATCHER J. W., Eds., *Complexity of Computer Computations*, p. 85–103, Plenum Press, New York, 1972.

[KUR 04] KURUVILA J., NAYAK A., STOJMENOVIĆ I., « Hop Count Optimal Position Based Packet Routing Algorithms for Ad Hoc Wireless Networks with a Realistic Physical Layer », *Proceedings of IEEE Mobile Ad hoc and Sensor Systems (MASS)*, Fort Lauderdale, FL, USA, 2004.

[MAR 04] MAROTI M., KUSY B., SIMON G., LEDECZI A., « The flooding time synchronization protocol », *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, USA, 2004, p. 39–49.

[QAY 02] QAYYUM A., VIENNOT L., A.LAOUITI, « Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks », *Proceedings of Annual Hawaii International Conference on System Sciences (HICSS)*, Hawaii, HI, USA, 2002.

[TIA 05] TIAN D., GEORGANAS N. D., « Connectivity maintenance and coverage preservation in wireless sensor networks », *AdHoc Networks Journal (Elsevier Science)*, , 2005, p. 744–761.

[XIN 05] XING G., WANG X., ZHANG Y., LU C., PLESS R., GILL C., « Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks », *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, n° 1, 2005, p. 36–72, ACM Press, New York, NY, USA.

[YEN 07] YENER B., MAGDON-ISMAIL M., SIVRIKAYA F., « Joint problem of power optimal connectivity and coverage in wireless sensor networks », *Wireless Networks*, vol. 13, n° 4, 2007, p. 537–550, Kluwer Academic Publishers.

[ZHA 05] ZHANG H., HOU J. C., « Maintaining Sensing Coverage and Connectivity in Large Sensor Networks », *Ad Hoc and Sensor Wireless Networks journal (AHSWN)*, vol. 1, 2005, p. 89–123.

Supprimer le protocole Neighbor Discovery dans les réseaux de capteurs

Laurent Toutain[*] [1] — **Kevin Perros**[*] [2] — **Joongsoo Lee**[**] [2]

[*] *ENST Bretagne,*
CS 17607, Cesson Sévigné 35576 Cedex - France,
Tel. : +33-2-99-12-70-52, Fax : +33-2-99-12-70-30
{Laurent.Toutain, Kevin.Perros}@enst-bretagne.fr

[**] *Information and Communications University,*
Munjiro 119, Yuseong-gu, Daejeon, Korea
Tel. : +82-42-866-6251, Fax : +82-42-866-6154
jslee@icu.ac.kr

RÉSUMÉ. *Cet article propose de nouveaux mécanismes pour réduire l'utilisation du protocole de découverte de voisins* Neighbor Discovery, *en particulier des* Router Advertisement, *dans un réseau de capteurs sans fil doté d'une pile IPv6. Les* Router Advertisement *sont des messages relativement longs, envoyés périodiquement sur le lien pour attester de la disponibilité du routeur. Les spécificités des réseaux de capteurs par rapport aux réseaux IPv6 classiques font qu'il n'est pas toujours nécessaire d'utiliser ces messages. Les mécanismes que nous proposons peuvent également être utilisés pour détecter les* handovers *dans les applications de mobilité.*

ABSTRACT. *This paper proposes new mechanisms to reduce the need of Neighbor Discovery, especially Router Advertisement in a Sensor Wireless Network implementing an IPv6 stack. Router Advertisement are relatively long messages, sent periodically on the link to show the availability of the router. Since Sensor Network Traffic differs from other form of traffic, it is not always necessary to send these messages on that kind of links. The proposed mechanisms can also be used for handover detection in mobility applications.*

MOTS-CLÉS : *6lowpan, Découverte de Voisins, Économie d'Énergie, Mobilité IPv6, Réseaux de Capteurs Sans Fil*

KEYWORDS: *6lowpan, Neigbor Discovery, Power Saving, IPv6 Mobility, Wireless Sensor Network*

1. Ce travail s'inscrit dans le projet "TINY 6" du programme "STIC-ASIE" soutenu par le MAEE, l'Institut Telecom et l'INRIA.
2. Ce travail s'inscrit dans le projet "MoMo" soutenu par le MIC (Ministry of Information and Communication) coréen et publié avec l'autorisation de la NIA (National Information Society Agency) de Corée.

1. Introduction

Le protocole *Neighbor Discovery* (NDP) [NAR 07] est utilisé par un équipement IPv6 pour dialoguer avec ses voisins dans le but d'établir la relation entre l'adresse IPv6 et l'adresse MAC. NDP est aussi largement utilisé pendant le démarrage de la machine pour auto-configurer les interfaces. Une grande majorité des équipements l'utilisent pour construire leur adresse globale et trouver le routeur par défaut. Le processus est le suivant : un équipement activant son interface réseau crée une adresse lien-local et, après avoir vérifié son unicité, envoie en *multicast* un message *Router Sollicitation* (RS) vers le groupe bien connu des routeurs du lien (`FF02::2`). Les routeurs répondent directement à l'équipement avec un message *Router Advertisement* (RA) qui peut contenir les préfixes utilisés sur le lien. L'équipement y concatène son identifiant d'interface (IID) pour produire son adresse globale. Il sélectionne aussi le routeur comme son routeur par défaut. De cette manière, l'équipement obtient une connectivité de niveau 3 et est capable d'émettre et de recevoir des paquets sur le réseau mondial. Les autres paramètres comme le résolveur DNS peuvent être appris grâce au protocole DHCPv6, ou, dans le cadre de standardisations récentes [JEO 07], peu être inclus dans les RA.

Les routeurs produisent également périodiquement des messages RA à l'adresse *multicast* bien connue des nœuds du réseau (`FF02::1`) pour :

– prouver aux équipements que le routeur est toujours actif ;

– indiquer sur quel réseau les équipements sont connectés.

Le dernier point est important en cas de mobilité IP car un nœud mobile peut utiliser le préfixe diffusé pour détecter un changement de réseau et produire un message d'association (*Binding Update*). Néanmoins, cette procédure est peu performante et conduit à une forte utilisation de la bande passante. Certaines mises en œuvre demandent une périodicité de quelques secondes [DAL 03]. La valeur la plus courante est de l'ordre d'une cinquantaine de secondes.

Dans les réseaux de capteurs, l'utilisation de messages périodiques peut conduire à de mauvaises performances, aussi bien en terme d'utilisation de la bande passante que pour la consommation électrique. Le but de ce papier est d'étudier les conséquences de la suppression des messages RA périodiques et d'étudier les conditions permettant la réduction des messages sollicités RS/RA tout en garantissant des communications de bout-en-bout avec les autres équipements connectés à l'Internet. Après une présentation de l'architecture IEEE 802.15.4 et du protocole 6lowpan de l'IETF, des modifications seront proposées pour réduire le besoin des messages RA périodiques et sollicités.

2. Architecture IEEE 802.15.4 et 6lowpan

IEEE 802.15.4 [IEE06] est une technologie réseau conçue pour réduire la consommation électrique. Elle est basée sur la méthode d'accès CSMA-CA avec un débit al-

lant de 20 à 250 kbit/s. Plusieurs réseaux (PAN : *Personal Area Network*) indépendants peuvent être déployés sur une même zone. Ils sont identifiés par un PANid. Chacun de ces réseaux possède un unique coordinateur (PC : *PAN Coordinator*) qui gère les attachements des capteurs au réseau, l'allocation d'adresses et optionnellement la bande passante en permettant d'écouler un trafic synchrone. Le PC assure aussi l'interconnexion avec d'autres technologies de réseaux. Dans notre cas, le PC sera mis en œuvre sur le routeur assurant l'interconnexion entre le réseau de capteurs et le réseau Internet.

Quatre types de trames sont définies par le standard : balise (*beacon*), données, acquittement et contrôle. Comme le montre la figure 1, elles commencent toutes par une séquence de synchronisation (préambule et SDF) suivi de la longueur de la trame sur 7 bits, permettant une taille maximale de 127 octets de données. Un champ contrôle donne la nature de la trame, indique la présence et la taille des adresses (16 bits ou 64 bits) et l'utilisation du PANid. Un champ séquence est utilisé pour identifier et acquitter les trames. A part les acquittements, toutes les autres trames contiennent un champ adresse composé du PANid et des adresses de la source et de la destination comme indiqué dans le champ contrôle. Seules les trames d'acquittement ne contiennent pas de données. Toutes les trames se terminent par un CRC.

Zigbee [Zig04] a été défini par le forum Zigbee pour permettre à des applications de fonctionner directement au dessus de la couche MAC. Zigbee offre des possibilités d'adressage, de routage et definit des applications. Zigbee n'est pas une architecture de bout-en-bout : les équipements connectés à l'Internet dialoguent uniquement avec le coordinateur Zigbee, qui interroge le capteur. L'approche de l'IETF avec le groupe 6lowpan est différente puisque le but est de rétablir la connectivité de bout-en-bout entre les équipements IEEE 802.15.4 et les autres équipements IPv6.

6lowpan [MON 07] est une couche d'adaptation utilisée pour transporter IPv6 au dessus d'un réseau de capteurs, et plus particulièrement sur les réseaux IEEE 802.15.4. Ce protocole n'autorise que des trames de 127 octets tandis que le standard IPv6 [DEE 98] impose une taille minimale de 1 280 octets. 6lowpan intègre un mécanisme de fragmentation pour s'adapter à cette contrainte. Il prévoit également la possibilité de compresser les en-têtes IPv6 pour éviter de transmettre les valeurs déjà connues ou contenues au niveau 2. 6lowpan définit le premier bit du champ données comme un discriminant qui indique comment les données sont structurées. Par exemple, la valeur `01 000001` indique que les octets suivants contiennent un paquet IPv6 dont l'en-tête n'est pas compressé. `01 000010` indique que l'en-tête est compressé, `10 xxxxxx` indique une encapsulation pour réseau maillé et `11 000xxx` et `11 100xxx` indiquent une fragmentation (respectivement le premier paquet et les paquets restants).

Actuellement, les réseaux de capteurs sont simples et centrés autour du coordinateur. Si l'IETF a presque finalisé la définition de la couche d'adaptation, ses travaux sur le routage n'en sont qu'à leur commencement.

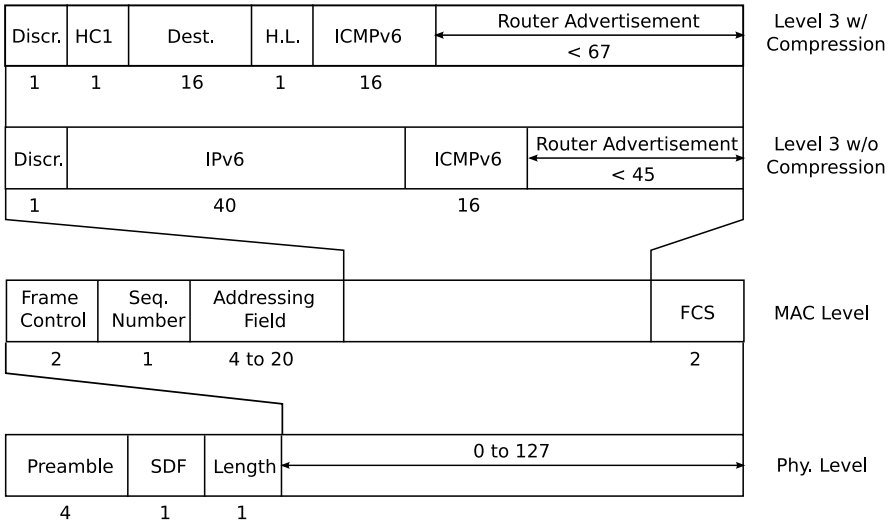


Figure 1. *Encapsulation des RA en IEEE 802.15.4*

2.1. *Compression de l'en-tête*

L'en-tête IPv6 est stable et les champs ont un comportement prévisible. Un discriminant a été défini pour les en-têtes compressés, il est suivit d'un *bitmap* indiquant quels champs doivent ou non transmis. Dans le meilleur des cas, seul le champ Nombre de Sauts (*Hop Limit*) est émis. Les adresses IPv6 source et destination peuvent aussi être compressées, si l'identifiant d'interface est dérivé de l'adresse MAC (16 ou 64 bits).

Quand une adresse de 16 octets est utilisée, son allocation est faite par le coordinateur, l'identifiant d'interface est construit en ajoutant l'adresse de niveau 2 et le PANid. Sinon l'adresse est dérivée des 64 bits de l'adresse MAC.

Dans le cas d'un message de *Neighbor Discovery*, l'adresse lien-local peut être complètement compressée. Par contre les adresses de *multicast* (tout les nœuds, tous les routeurs) ne peuvent pas l'être et les 16 octets doivent être transmis.

Les spécifications 6lowpan proposent de créer des adresses *multicast* de niveau 2 basées sur les adresses *multicast* IPv6. Dans ce cas, les derniers 13 bits de l'adresse IPv6 de *multicast* sont recopiés dans une adresse MAC de16 bits. Hui [HUI 07] propose d'étendre ce format pour y inclure la portée et le groupe. Si ce schéma était retenu, le champ adresse contiendrait l'adresse de la source (16 ou 64 bits) et l'adresse *multicast* (16 bits), ce qui aurait pour conséquence de compresser complètement l'entête. Les versions actuelles ne supportent pas cette possibilité, l'adresse de *broadcast* de 16 bits (`0xFFFF`) doit être utilisée et l'adresse de *multicast* doit être incluse dans le message.

2.2. *Compression des RA*

Quand un capteur entre dans un réseau 6lowpan, il doit construire son adresse IPv6 (soit en la demandant au coordinateur, soit en la construisant à partir de son adresse MAC-64 et en faisant un test pour s'assurer de son unicité). Comme expliqué précédemment, le nœud envoie un RS et reçoit un RA contenant le(s) préfixe(s) disponibles sur le lien et utilise le routeur ayant répondu comme routeur par défaut.

RS est un petit message contenant les 16 octets de l'en-tête ICMPv6 plus une option contenant l'adresse MAC de la source qui ne demande pas de compression particulière et qui est envoyée sur le groupe de *multicast* « tous les routeurs » (`FF02::2`). La norme 6lowpan ne prévoit pas d'adressage *multicast* niveau 2 dans le cas des topologies en étoile [MON 07]. Cette adresse est donc transformée au niveau 2 en adresse de diffusion généralisée, tous les nœuds du PAN reçoivent ce message et, à part le routeur, le rejettent.

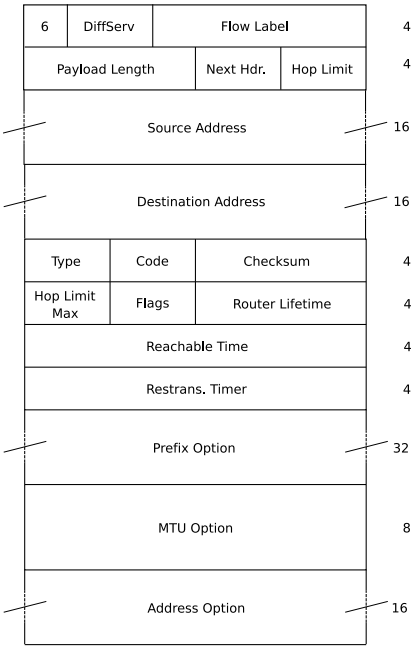


Figure 2. *Message* Router Advertisement *Complet*

La figure 2 donne le format d'un message RA pour un réseau IEEE 802.15.4 quand l'adresse MAC est codée sur 64 bits. Ce message contient :

– 40 octets pour l'en-tête IPv6 ;

– 16 octets pour l'en-tête IPv6, contenant des paramètres commun au réseau comme la valeur maximale du champ nombre de sauts, la durée en seconde pendant laquelle le routeur assurera les fonctions de routeur par défaut, des drapeaux

pour connaître l'articulation avec DHCPv6 et deux temporisateurs pour la résolution d'adresses ;

– L'en-tête ICMPv6 est suivi d'options : 32 octets pour l'option contenant le préfixe et sa durée de validité, 8 pour la valeur de la MTU recommandée sur le lien et 16 ou 8 octets pour l'adresse physique du routeur. Cette dernière option peut être omise, mais elle permet aux nœuds de trouver facilement l'adresse physique du routeur par défaut.

Comme indiqué figure 1, dans le pire des cas (les adresses sont codées sur 64 bits), seuls 45 octets sont disponibles dans une trame IEEE 802.15.4 pour transporter les options. L'option « adresse » ne peut pas être incluse, sauf si on a recourt à la fragmentation. La compression d'en-tête réduit l'en-tête IPv6 de 40 à 18 octets (HC1 + Nombre de Sauts + adresse destination) laissant assez d'espace pour toutes les options du RA. Bien entendu ce calcul n'est valable que lorsqu'un seul préfixe est transmis. Si plusieurs préfixes existent, la fragmentation sera obligatoire.

3. Suppression des RA sollicités

Si l'échange RS/RA peut être évité, cela réduira la consommation en énergie du système, principalement parce que les requêtes RS sont diffusées vers tous les équipements. Dans certain cas, un capteur initiant une session avec le monde extérieur n'a pas besoin de connaître sa propre adresse IPv6. Cette adresse peut être ajoutée par le routeur de bordure. Quand le destinataire répond, le paquet est retourné au routeur de bordure grâce au préfixe et le routeur compresse l'en-tête IPv6 et en extrait l'adresse destination qu'il place dans la trame et l'information est transmise au capteur.

Ce schéma est très similaire à celui utilisé quand, en IPv4, un adressage privé est mis en place. L'adresse privée correspond à l'IID et l'adresse publique correspond au préfixe. La différence majeure comparée à la traduction d'adresse est que si le capteur veut réellement connaître son adresse publique, il peut envoyer un message RS sur le réseau et attendre la réponse du routeur.

Hui [HUI 07] définit un nouveau discriminant pour la compression des en-têtes appelé HC1g pour les adresses globales. Ce discriminant fonctionne de la même manière que celui standardisé par 6lowpan, mais au lieu d'ajouter le prefixe lien-local, le prefixe global du lien est inséré si un paquet IPv6 complet doit être construit. Dans notre cas, ce nouveau discriminant n'est pas nécessaire puisque les paquets doivent être transmis à l'extérieur du réseau de capteurs, l'adresse de destination contient déjà un préfixe global et son identifiant (incompressible puisqu'imprévisible). Un routeur qui reçoit ce type de paquet ajoute le préfixe du lien et l'adresse physique et la source.

Si les RA sont supprimés, le routeur par défaut doit être localisé par les capteurs. Quand le réseau de capteurs a une topologie en étoile, toutes les communications sont organisées autour du coordinateur, donc le routeur est intrinsèquement découvert. Pour les autres topologies, un protocole de routage de niveau 2 est nécessaire pour acheminer les paquets jusqu'au routeur. Dans cette optique, nous proposons d'identifier

le routeur par défaut par une adresse *anycast* bien connue. Cela veut dire que tout équipement ayant les fonctions de routeur par défaut doit être configuré avec cette adresse et que le protocole de routage à l'intérieur du réseau de capteurs doit relayer les trames vers cette adresse. Les capteurs seront configurés avec cette adresse par défaut dans leur table de routage IPv6 et relayeront tous les paquets vers cette adresse. Cette adresse *anycast* ne sera pas utilisée si un RA a fourni une autre valeur. Nous n'étudions pas le routage *anycast* niveau 2 pour les topologies en maillage dans cet article.

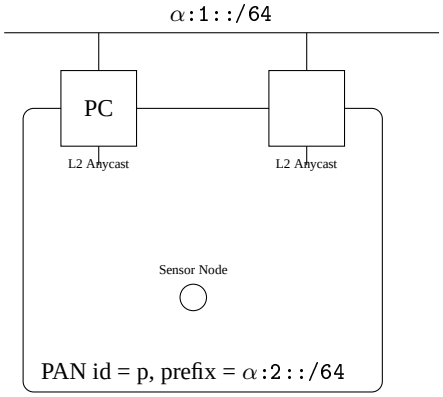


Figure 3. *Adresse* anycast *de niveau 2*

Dans le cas de plusieurs attachement au réseau de capteurs, comme représenté figure 3, plusieurs routeurs par défaut (mais un seul PC) seront actifs sur le réseau. La trame sera routée au niveau 2 vers le routeur le plus proche, réduisant ainsi la consommation du système global en limitant le nombre de sauts. Avec la découverte standard du routeur par défaut, le premier routeur répondant est choisi, le chemin peut donc être plus long, et ne changera pas, même si le capteur se déplace.

Les RA periodiques peuvent être utilisés pour détecter la mort d'un routeur. Dans le cas d'une adresse *anycast* de niveau 2, il n'est pas nécessaire de détecter la mort du routeur, puisque si un autre existe, les paquets seront redirigés vers lui. Certaines applications peuvent vouloir savoir si le routeur par défaut est toujours actif. Une solution possible serait d'utiliser les trames d'acquittement prévues dans le standard IEEE 802.15.4.

Une contrainte à cette approche peut apparaître quand les messages sont fragmentés, si le routage est instable, ou si l'équipement se déplace : certains fragments peuvent être reçus par un routeur et d'autres par un autre, rendant le ré-assemblage impossible. Néanmoins, vues la petite taille et la sporadicité du trafic, le risque est faible. Le capteur ayant un trafic avec de grands paquets peut apprendre une route par défaut grâce à la procédure standard RS/RA.

4. Gestion de la mobilité

Les RA périodiques peuvent être utilisés lors de situations de mobilité pour détecter un *handover* [DAL 03]. Si ceux-ci ne sont plus transmis, un capteur pourra prendre plus de temps pour détecter un déplacement. L'information sur le préfixe doit être résumée ailleurs. Nous proposons d'utiliser le PANid pour informer des modifications de la valeur du préfixe. Le PANid etant sur 16 bits, l'algorithme doit avoir les propriétés suivantes :

– si le capteur reste dans le même domaine (i.e. identifé par le même préfixe global) mais se déplace d'un sous-réseau vers un autre, le PANid doit changer ;

– si le capteur se déplace d'un domaine à un autre, le risque de ne pas détecter ce mouvement doit être faible.

Un algorithme comme $PANid = hash(Préfixe\ Global) + SID$, ou plus simplement l'algorithme bien connu de calcul de somme de contrôle des en-têtes IP (addition en complément à 1 des mots de 16 bits du préfixe), répondent à ces propriétés. Aucune hypothèse n'est faite sur la longueur du préfixe global, s'il est plus grand ces propriétés sont respectées.

Les propriétés du PANid ainsi obtenu sont les suivantes : si le capteur quitte un domaine pour entrer dans un autre, alors le préfixe global IPv6 et le SID changent, il y a alors une probabilité de $1/65536$ que le changement ne soit pas détecté.

Comme le coordinateur est aussi le routeur par défaut, quand un préfixe est alloué au PC, le PANid est calculé et annoncé sur le réseau de capteurs. En cas d'attachements multiples, comme le décrit la figure 3, un changement de comportement par rapport aux spécifications de la norme IEEE 802.15.4 doit être fait. Le standard stipule que chaque PC doit posséder un PANid unique : en cas de conflit entre deux PC, l'un des deux doit changer la valeur de son PANid. Nous désirons au contraire qu'un PANid soit associé à chaque préfixe. Nous proposons qu'en cas de conflit un seul routeur garde le rôle de PC, les autres pouvant servir de PC de secours.

5. Conclusion

Nous avons présenté une meilleure intégration de IPv6 dans IEEE 802.15.4 pour réduire l'utilisation du protocole *Neighbor Discovery* dans les réseaux de capteurs sans fil. Cette approche permet à l'application de choisir le comportement du réseau. Certaines applications simples n'aurons pas besoin de configurer leur niveau 3 pour pouvoir émettre sur le réseau : le PC se chargera dans ce cas de construire les en-têtes niveau 3. Dans ce cas, on diminue grandement l'énergie dépensée pour la configuration et la gestion du réseau. Les capteurs qui voudraient une connectivité IPv6 complète de bout-en-bout peuvent tout de même utiliser *Neighbor Discovery* pour cela : nous avons montré que même dans ce cas, l'envoi de messages périodiques n'est pas nécessaire. Les capteurs peuvent également bénéficier de cette approche pour détecter les *handover* en écoutant les trames IEEE 802.15.4 plutôt que les messages RA.

6. Bibliographie

[DAL 03] DALEY G., PENTLAND B., NELSON R., « Effects of Fast Router Advertisement on Mobile IPv6 Handovers », *Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC 03)*, Kemer - Antalya, Turkey, July 2003, IEEE, p. 1530–1346.

[DEE 98] DEERING S., HINDEN R., « Internet Protocol, Version 6 (IPv6) Specification. », IETF Request for Comments n° 2460, December 1998, Internet Engineering Task Force.

[HUI 07] HUI J., CULLER D., « Stateless IPv6 Header Compression for Globally Routable Packets in 6LoWPAN Subnetworks », Internet Draft n° draft-hui-6lowpan-hc1g-00.txt, June 2007, Internet Engineering Task Force.

[IEE06] IEEE, « Standard 802.15.4-2006, Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). », june 2006.

[JEO 07] JEONG J., PARK S., BELOEIL L., MADANAPALLI S., « IPv6 Router Advertisement Option for DNS Configuration », IETF Request for Comments n° 5006, September 2007, Internet Engineering Task Force.

[MON 07] MONTENEGRO G., KUSHALNAGAR N., HUI J., CULLER. D., « Transmission of IPv6 Packets over IEEE 802.15.4 Networks. », IETF Request for Comments n° 4944, September 2007, Internet Engineering Task Force.

[NAR 07] NARTEN T., NORDMARK E., SIMPSON W., SOLIMAN H., « Neighbor Discovery for IP version 6 (IPv6). », IETF Request for Comments n° 4861, September 2007, Internet Engineering Task Force.

[Zig04] « ZigBee Specification Version 1.0. », december 2004.

Localisation collaborative pour réseaux de capteurs

Karel Heurtefeux— Fabrice Valois

ARES INRIA / CITI, INSA-Lyon, , France
Bâtiment Léonard de Vinci
21, avenue Jean Capelle
F-69621 Villeurbanne Cedex
{karel.heurtefeux,fabrice.valois}@insa-lyon.fr

RÉSUMÉ. *L'utilisation de mécanismes de localisation dans les réseaux de capteurs est essentiel à la fois pour les protocoles de communication (routage géographique) que pour certaines applications (suivi de véhicules). Dans cet article nous proposons un nouvel algorithme localisé sans GPS, sans ancre et sans spécificité matérielle particulière. Chaque nœud va construire une table de distance qualitative basée sur la collaboration avec son voisinage,* i.e. *basé sur les informations reçues de ses voisins 1 et 2 sauts grâce à un échange de paquets* `hello`*. Ainsi cet algorithme va permettre à chaque nœud du réseau de classer ses voisins en trois catégories :* très proche*,* proche *et* éloigné*. Nous avons étudié notre solution sur une topologie particulière, la grille, puis nous évaluons ses performances sur des topologies aléatoires.*

ABSTRACT. *The use of localization mechanism is essential in wireless sensor networks either for communication protocols (geographic routing protocol) or for applications (vehicle tracking). The goal of localization mechanism is to determine either precisely or coarsely the node location using either a global reference (GPS) or a locale one. In this work, we introduce a new localized algorithm which classified the proximity of the neighborhood for a node. This qualitative localization does not use any anchor or dedicated hardware like a GPS. Each node builds a Qualitative Distance Table according to the 2-hop neighborhood informations. Thus, the algorithm allows to determine coarsely the location of the neighbors which are classified as* very close*,* close *or* far*. The algorithm is analyzed both on a regular topology (grid) and a random one.*

MOTS-CLÉS : *contrôle de topologie, localisation, réseaux de capteurs, sans GPS, distance qualitative.*

KEYWORDS: *toplogy control, localisation, sensor networks, GPS-free, qualitative distance.*

1. Introduction

De nombreuses applications pour les réseaux de capteurs comme la surveillance de feu de forêt, le suivi de véhicule, etc... ont besoin d'avoir une information géographique pour fonctionner efficacement. Les protocoles de routage géographique ou orienté position vont pouvoir fonctionner sans le coûteux mécanisme de découverte de route proactive et ainsi économiser de l'énergie et améliorer le taux d'acheminement. De plus, dans les protocoles de contrôle de topologie, où chaque capteur doit ajuster sa puissance de transmission pour minimiser sa consommation énergétique, les algorithmes ont le plus souvent besoin d'information sur la position des voisins.

Le GPS [HOF 01] résout le problème de la localisation dans les environnements extérieurs. Cependant, pour les larges réseaux de capteurs où les nœuds doivent être de très petite taille, peu consommateur et peu onéreux, équiper chaque unité avec une puce GPS est trop coûteux.

Dans ce papier nous proposons un algorithme qui va permettre à chaque nœud de localiser *qualitativement* ses voisins en utilisant uniquement des informations locales. Notre objectif est de montrer qu'il existe une possibilité d'obtenir des informations assez précises sur la position des nœuds dans un réseau de capteurs sans s'appuyer ni sur un sous ensemble de nœuds connaissant leur position (ancres), ni sur des équipements spéciaux. L'algorithme déployé sur un nœud utilise seulement des informations obtenues par l'échange de table de voisinage à l'aide de paquets `hello` classique pour calculer un **indice de proximité** pour chaque voisin à un saut. Nous montrons que, malgré une légère imprécision inévitable, l'algorithme reste assez fiable et presque parfait sur les topologies particulières (grille). La figure 1 illustre le résultat de l'algorithme : les voisins à un saut du nœud étudié sont classés en 3 catégories : très proche, proche et loin.

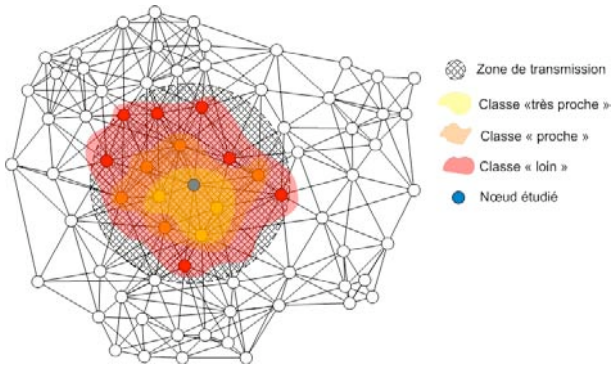


Figure 1. *Exemple d'une application de l'algorithme pour un nœud*

Ce papier est organisé de la façon suivante : dans le chapitre 2 nous présentons différentes techniques de localisation existantes. L'algorithme de localisation qualitative

est présenté chapitre 3. Les hypothèses que nous avons prises ainsi que les résultats obtenus sont présentés dans le chapitre 4. Dans le chapitre 5, nous concluons ce travail avec une présentation des travaux futurs.

2. État de l'art

De nombreuses techniques sont proposées pour permettre aux nœuds d'estimer leur position. Nous pouvons distinguer deux types de stratégie de localisation : les localisations fines et les localisations approximatives. Les localisations fines détermineront de manière précise les coordonnées du nœud dans le réseau tandis que les localisations approximatives, ou grossières, spécifieront une surface ou des coordonnées virtuelles, etc...

2.1. *Localisations fines*

L'utilisation du GPS permet d'éviter le problème de la localisation en extérieur. Cependant son coût, à la fois économique et énergétique, pose soucis. Quelques papiers contournent le problème et proposent l'utilisation d'ancres : seulement certains nœuds connaissent leur position précise et permettent autres nœuds du réseau, par triangulation ou multilatération de connaître la leur. Pour cela, plusieurs solutions sont proposées :

- La mesure de la puissance du signal reçue. Cette mesure est irréaliste car le signal radio est perturbé par l'environnement et varie grandement au cours du temps,

- ToA (Time of Arrival) [CAP 01] mais ce mécanisme nécessite une horloge synchronisée entre les nœuds,

- TDoA (Time difference of Arrival) [WAR 97], [NAW 07] : deux signaux de différentes natures sont utilisés (ultra-sons et radio par exemple),

- AoA (Angle of Arrival) [NIC 03], [AKC 06] : Cette méthode permet de déterminer la direction de propagation d'une onde radio reçue par un ensemble d'antennes,

- Combinaison de TDoA et de AoA [MAG 07] pour améliorer la précision et étendre des environnements 2D à ceux 3D la méthode de Capkun [CAP 01].

Tous ces protocoles ne prennent pas en compte la consommation énergétique et le coût des nœuds avec comme hypothèses des capteurs suréquipés. En outre, le système des ancres n'évite pas le problème de la localisation mais ne fait que le réduire à un sous-ensemble des nœuds du réseau. De plus il fait apparaître d'autres problèmes comme le placement optimal de ces ancres dans le réseau, pour permettre une meilleure localisation des nœuds [BEN 07], [DES 07], [SAA 06a], [SAA 06b].

2.2. *Localisations approximatives*

D'autres stratégies consistent à trouver des coordonnées approximatives. Selon l'application, une localisation approximative des capteurs peut être un bon compromis et dans ce cas plusieurs approches sont possibles :

- Le système des badges actifs [HOP 93] : chaque nœud est marqué et transmet un paquet hello périodique toutes les 10 secondes. Le signal est reçu par des capteurs placés à des endroits précis et fixes à l'intérieur d'un bâtiment et qui vont relayer l'information jusqu'au puits.

- L'algorithme d'estimation de position [HU 04] calcule des probabilités de distribution des positions possibles des nœuds. En fonction de la position précédente des nœuds et des observations provenant d'ancres déployées dans le réseau, l'algorithme est capable de filtrer les positions impossibles.

- Le système de coordonnées virtuelles [CAO 06] : chaque nœud détermine sa distance aux ancres en nombre de sauts et détermine ainsi ses propres coordonnées dans un système dont la dimension dépend du nombre d'ancres.

Ces protocoles sont peu adaptés aux réseaux de capteurs car ils requièrent soit des ancres fixées à une architecture fixe soit un calcul centralisé.

3. Algorithme de localisation qualitative

Il faut garder à l'esprit que le but de notre algorithme est de permettre à chaque nœud de déterminer grossièrement la position de ses voisins en utilisant uniquement des informations locales. Ces informations locales proviennent de paquets hello échangés périodiquement entre les voisins à 1 saut. La position qualitative d'un voisin peut être *très proche*, *proche* ou *loin*. Ainsi, cette position imprécise peut être utilisée pour construire un protocole de routage unicast efficace dans un environnement sans fil avec un haut niveau d'interférences : choisir les voisins considérés comme *très proches* permet de choisir les nœuds avec un rapport signal sur bruit important. Des applications au contrôle de topologie et aux coordonnées virtuelles pour le routage sont également possibles.

Le nœud A calcule l'indice de proximité d'un voisin B de la façon suivante :

$$PI_A(B) = (|V(a) \cap V(b)|) - \frac{max(|V(a)|, |V(b)|)}{2}$$

avec $V(a)$ le voisinage de A.

L'idée générale est d'attribuer un indice fort aux voisins ayant à la fois un voisinage commun important et un voisinage distinct faible. L'indice de proximité va ainsi évaluer la similarité des voisinages. Ainsi, nous prenons en compte dans l'algorithme le rapport entre le nombre de voisins communs et le nombre de voisins disjoints. Il

semble en effet logique que des voisins proches géographiquement aient un voisinage fortement semblable alors que deux voisins éloignés vont avoir un voisinage assez dissemblable. Ainsi l'indice de proximité va représenter la distance **qualitative** d'un voisin. Nous allons voir que cette proximité qualitative va de paire avec la proximité géographique dans le cas des réseaux de capteurs dense. Ce mécanisme permet d'établir trois classe distinctes parmi les voisins : la classe *très proche* (ou 1), la classe *proche* (ou 2) et la classe *loin* (ou 3). Nous déterminons les classes des nœuds de la façon suivante :

Soit $PI(x)$ l'indice de proximité du voisin x :

$$inter = \frac{|max(PI(x_i)) - min(PI(x_i))|}{3}$$

$$classe_x = \begin{cases} 1 & si \quad PI(x) \geq max(PI(x_i)) - inter \\ 2 & si \quad max(PI(x_i)) - inter > PI(x) \geq max(PI(x_i)) - 2.inter \\ 3 & si \quad PI(x) < max(PI(x_i)) - 2.inter \end{cases}$$

Chaque nœud du réseau calcule l'indice de proximité de chacun de ses voisins en fonction des informations reçues de son voisinage à 1 saut. Chaque nœud maintient une table de voisinage à 1 et 2 sauts et diffuse celle à 1 saut avec des paquets `hello` périodiques. La figure 2 et le tableau 3 montre une application de l'algorithme sur un nœud particulier. Le nœud 27 ordonne ses voisins en 3 classes de proximité. Nous pouvons voir en détails les valeurs trouvées par l'algorithme dans le tableau 3. Celui-ci permet également de mettre en parallèle ces valeurs avec celles déterminées par la distance euclidiennes. Notez que sur cet exemple le réseau est relativement peu dense.

Le protocole est peu consommateur en énergie car il utilise seulement des informations nécessaires à beaucoup d'autres protocoles déployés dans les réseaux de capteurs : auto-organisation (CDS-règle-k [WU 99], CDS-MIS [WAN 02],...) et protocoles de routages pro-actifs (OLSR,... [CLA 03]). De plus, si le réseau est peu dynamique (faible mobilité, peu de naissance ou de mort de nœuds dans le réseau [HEU 07]) cet échange de paquets peut être réduit voire limité simplement à la phase de déploiement du réseau.

4. Résultats

Tous les résultats proviennent de simulations. Nous avons utilisé un simulateur à événements discrets Java in Simulation Time (JiST) et le module Scalable Wireless Ad hoc Network Simulator (SWANS) [BAR 05]. La topologie du réseau de capteurs est modélisée comme un graphe de disques unitaires (UDG) avec une couche MAC de type CSMA-CA. La cardinalité du réseau varie entre 50 et 700 nœuds, répartis de façon aléatoire et uniforme sur la surface de simulation excepté lors de l'étude sur la topologie particulière où les nœuds ont été placés de façon régulière sur une grille.

6
.

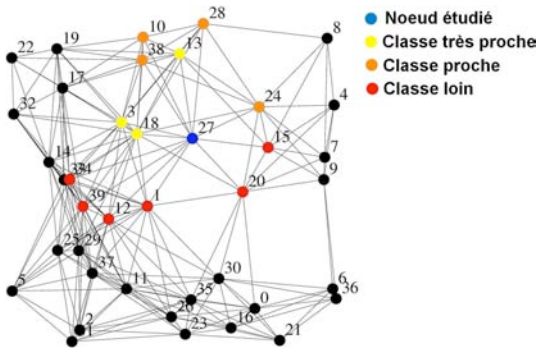


Figure 2. *Exemple de localisation qualitative calculé par le nœud 27.*

Voisins	indice de proximité	distance euclidienne	classe de proximité	classe réelle
18	2.0	50.067	very close	very close
3	1.0	65, 18	very close	very close
13	0.5	77, 01	very close	close
38	−0.5	83, 66	close	far
28	−0.5	103, 76	close	far
24	−0.5	66, 20	close	very close
10	−1.5	101, 18	close	far
1	−2.0	73, 09	far	close
20	−2.5	65, 96	far	very close
39	−3.0	115, 62	far	far
34	−3.0	115, 98	far	far
15	−3.5	68, 28	far	very close
12	−4.0	104, 40	far	far

Figure 3. *Comparaison de la localisation pour les voisins du nœud 27 obtenue avec l'algorithme de localisation qualitative et en utilisant la position réelle.*

Chaque nœud est immobile. La puissance de transmission est utilisée pour contrôler le degré moyen des nœuds du réseau. L'objectif est d'étudier notre protocole et d'observer son efficacité dans le classement des nœuds voisins.

4.1. *Comportement du protocole de localisation qualitative sur une topologie régulière*

Nous simulons un réseau de 100 capteurs déployés de façon uniforme et régulière sur une grille de 10x10. Ensuite, nous augmentons la puissance de transmission de chacun des capteurs et nous observons comment réagit le protocole de localisation

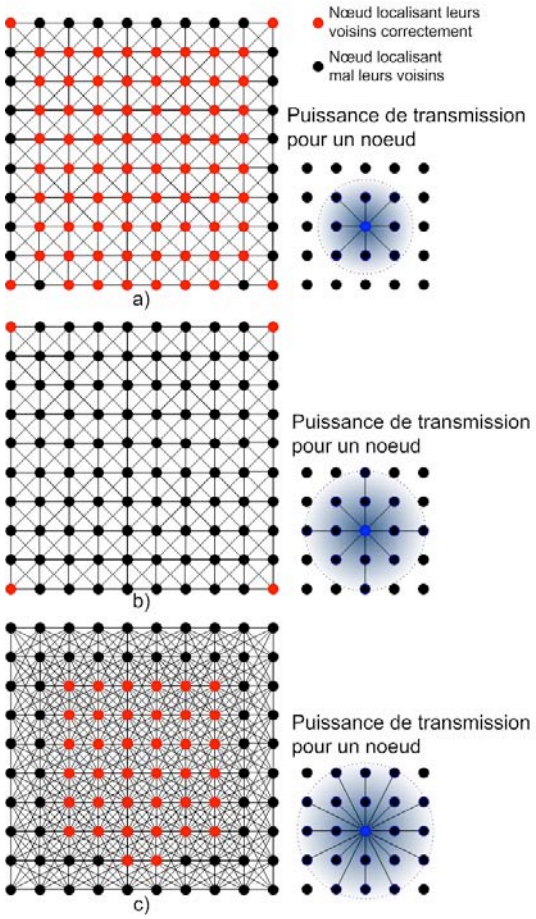


Figure 4. *Algorithme déployé sur une grille*

qualitative. Parfois le voisinage n'est pas représentatif de la régularité de l'ensemble du réseau (cf. Fig 4). Dans ce cas (voir scénario b) Fig. 4) ou quand les nœuds se trouvent sur les bords, l'algorithme ne parvient pas à ordonner correctement les voisins des deux premières classes de proximité à cause des incohérences dans le voisinage. Pour les autres topologies, les différentes classes sont déterminées sans erreur et l'indice de proximité est identique pour des voisins éloigné de la même distance géographique. Nous pouvons conclure que quand la topologie et le voisinage sont uniformes et réguliers, la localisation qualitative est très efficace.

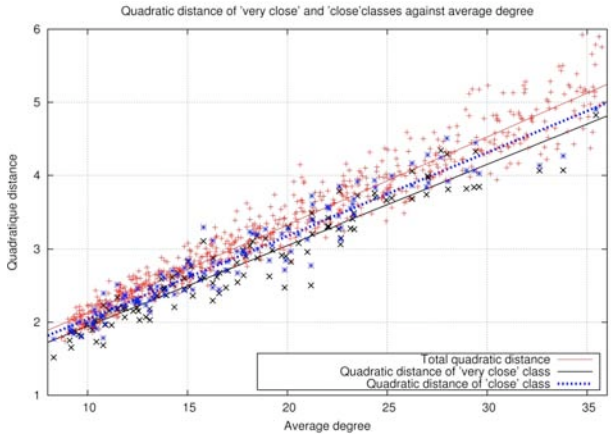


Figure 5. *Distance quadratique par classe*

4.2. *Comportement du protocole de localisation qualitative sur des topologies aléatoires*

Cependant, les réseaux de capteurs sont, la plupart du temps, déployés de façon aléatoire. Afin de mesurer la précision de l'algorithme dans un environnement plus réaliste, nous déployons, selon une distribution aléatoire et uniforme, 100 nœuds dont nous faisons varier la puissance de transmission pour augmenter le degré. Nous calculons ensuite la distance quadratique entre la liste des voisins de chacun des nœuds classée selon la position réelle des nœuds (GPS) et celle classée par l'algorithme.

Soit deux listes v et w dans R^n tel que : $v = (v_1, v_2, ..., v_n)$, $w = (w_1, w_2, ..., w_n)$. La distance quadratique dq est :

$$dq = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - w_i)^2}$$

Dans cet article nous étudions la distance quadratique de l'algorithme pour les classe *très proche* et *proche* et pour toutes les classes. Nous observons que la distance quadratique augmente mais de façon plus lente que le degré moyen. Quand le degré moyen augmente le nombre de voisins à classer pour chaque nœud augmente. Si la distance quadratique reste basse, cela signifie que la précision augmente. Ce phénomène s'explique par le grand nombre d'information reçu quand le voisinage est important et donc une plus grande fiabilité dans l'indice de proximité. Les différentes classes varient de la même façon. En outre, nous pouvons noter une plus faible augmentation pour les deux premières classe : proche et très proche.

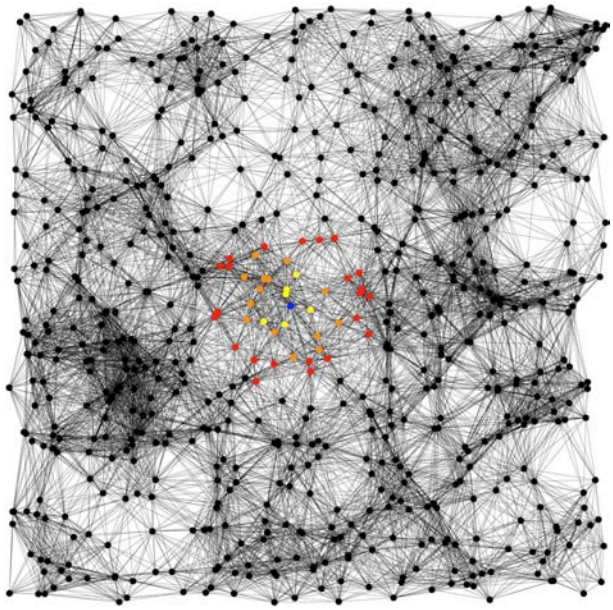


Figure 6. *Les 3 classes de l'algorithme*

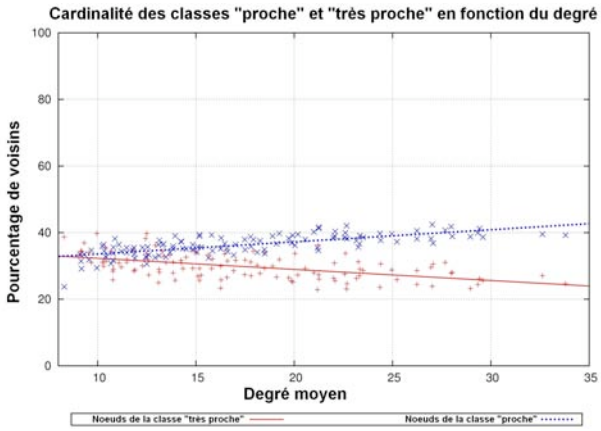


Figure 7. *Évolution de la cardinalité des classes de proximité en fonction du degré moyen*

Dans le cas d'une topologie dense (700 nœuds, degré moyen : 40), la localisation est très efficace. Nous pouvons voir le classement en trois classe, par le nœud 27 de son voisinage figure 6. Les nœuds jaunes appartiennent à la classe *très proche*, les oranges à la classe *proche* et les rouges à la classe *loin*.

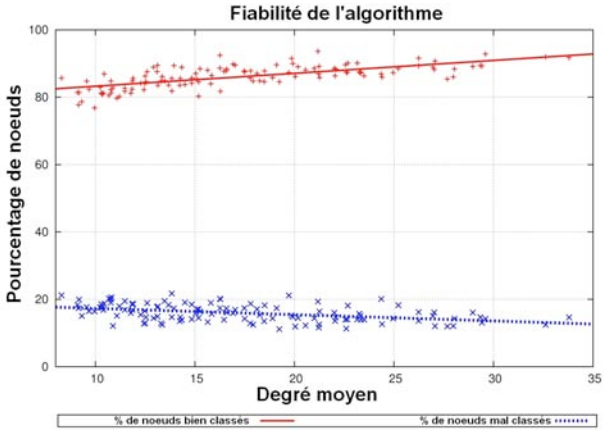


Figure 8. *Évolution de la fiabilité de l'algorithme en fonction du degré moyen*

Chaque nœud alloue une classe à ses voisins en fonction de son indice de proximité. Comment évoluent ces classes quand le degré moyen augmente ? La classe *très proche* va-t'elle croître proportionnellement au nombre de voisin ? Nous avons vu que la distance quadratique augmente légèrement quand le degré moyen augmentait. Cependant cette métrique est très sensible à la taille des listes évaluées. Ainsi nous étudions le pourcentage moyen de nœuds sélectionnés dans la classe *très proche* et *proche* (Fig. 7). Nous pouvons souligner que le pourcentage des nœuds de la classe *très proche* diminue alors que, dans le même temps, la classe *proche* augmente. La cardinalité de classe *loin* reste constante. Cela indique que plus la densité est importante et plus l'indice de proximité est capable de distinguer les nœuds réellement proches.

Quand l'algorithme détermine une classe de distance pour un voisin, il est primordial de savoir si ce nœud sélectionné comme proche ou très proche, l'est effectivement dans le monde réel. Pour répondre à cette interrogation, nous avons déterminé le nombre de nœuds appartenant à la classe *proche* et *très proche* sélectionnés par l'algorithme appartenant effectivement à ces classes géographiquement. Ensuite nous regardons le nombre de nœuds réellement dans ces deux classes et non sélectionnés par l'algorithme. Nous avons ainsi la fiabilité de l'algorithme en répondant à ces deux questions : quelle probabilité qu'un voisin classé comme proche ou très proche le soit effectivement et quelle probabilité qu'un nœud proche ou très proche soit manqué par l'algorithme. Nous observons une fiabilité de plus de 80% même pour des topologies de faible degré (Fig 8).

5. Conclusion et perspectives

Dans ce travail, nous proposons un algorithme de localisation des voisins utilisant uniquement des informations locales. Notre solution fonctionne sans GPS, ni ancre ou

équipement spécial. Basé sur des informations provenant de son voisinage à 1 saut, un nœud peut qualitativement classer son voisinage en des groupes de nœuds *très proches*, *proches* ou *loins*. Nous avons étudié le comportement de notre algorithme sur une topologie régulière et sur des topologies aléatoires. La distance quadratique est calculée pour mettre en évidence l'efficacité de la classification obtenue. Nos premières simulations nous donnent des résultats très encourageants en termes d'énergie et permet d'envisager un algorithme de contrôle de topologie sans ancre ni GPS. Ensuite nous appliquerons l'algorithme de localisation qualitative pour construire un protocole de routage basé sur des coordonnées virtuelles, ainsi qu'un protocole de routage unicast adapté aux réseaux de capteurs avec interférences.

6. Bibliographie

[AKC 06] AKCAN H., KRIAKOV V., BRÖNNIMANN H., DELIS A., « GPS-Free node localization in mobile wireless sensor networks », *ACM MobiDE*, New York, NY, USA, 2006, p. 35–42.

[BAR 05] BARR R., HAAS Z., VAN RENESSE R., « *Scalable Wireless Ad hoc Network Simulation* », chapitre 19, p. 297–311, CRC Press, August 2005.

[BEN 07] BENBADIS F., OBRACZKA K., CORTÈS J., BRANDWAJN A., « Exploring landmark placement strategies for self-localization in wireless sensor networks », *IEEE PIMRC*, Athens, Greece, September 2007.

[CAO 06] CAO Q., ABDELZAHER T., « Scalable logical coordinates framework for routing in wireless sensor networks », *ACM Transactions on Sensor Networks*, vol. 2, n° 4, 2006, p. 557 - 593.

[CAP 01] CAPKUN S., HAMDI M., HUBAUX J.-P., « GPS-Free Positioning in Mobile ad-hoc Networks », *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, vol. Volume 9, 2001, page 9008, IEEE Computer Society, Location.

[CLA 03] CLAUSEN T., JACQUET P., « Optimized Link State Routing Protocol (OLSR) », october 2003, RFC 3626.

[DES 07] DESAI J., TURELI U., « Evaluating performance of various localization algorithms in wireless and sensor networks », *IEEE PIMRC*, Athens, Greece, September 2007.

[HEU 07] HEURTEFEUX K., VALOIS F., « Topology Control Algorithms : a qualitative study during the sensor networks life », *3rd International Workshop on Localized Communication and Topology Protocols for Ad hoc Networks (LOCAN'07), in conjunction with MASS*, Pisa, Italy, October 2007.

[HOF 01] HOFMANN-WELLENHOF B., LICHTENEEGER H., COLLINS J., *Global Positioning System : Theory and Practice*, Springer Wien New York, 2001.

[HOP 93] HOPPER A., HARTER A., BLACKIE T., « The active badge system (abstract) », *CHI '93 : Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, New York, NY, USA, 1993, ACM Press, p. 533–534.

[HU 04] HU L., EVANS D., « Localization for mobile sensor networks », *ACM MobiCom*, New York, NY, USA, 2004, p. 45–57.

[MAG 07] MAGNANI A., LEUNG K., « Self-Organized, Scalable GPS-Free Localization of Wireless Sensors », *IEEE WCNC*, Hong Kong, China, 2007, p. 3798–3803.

[NAW 07] NAWAZ S., JHA S., « Collaborative locatization for wireless sensor networks », *IEEE PIMRC*, Athens, Greece, September 2007.

[NIC 03] NICULESCU D., NATH B., « Ad Hoc Positioning System (APS) using AOA », *IEEE INFOCOM*, San Francisco, USA, 2003.

[SAA 06a] SAAD C., « Une nouvelle méthode de positionnement dans les réseaux de capteurs », *AlgoTel*, Trégastel, France, 2006.

[SAA 06b] SAAD C., BENSLIMANE A., KÖNIG J.-C., « MuR : A Distributed Preliminary Method For Location Techniques in Sensor Networks », *IEEE WiMob*, Montréal, Canada, 2006, p. 61–68.

[WAN 02] WAN P., ALZOUBI K., FRIEDER O., « Distributed construction of connected dominating set in wireless ad hoc networks », *INFOCOM*, New York, NY, USA, 2002.

[WAR 97] WARD A., A.JONES, HOPPER A., « A new location technique for the active office », *IEEE Personal Communications*, vol. 4, 1997, p. 42-47.

[WU 99] WU J., LI H., « On calculating connected dominating set for efficient routing in ad hoc wireless networks », *ACM DIALM*, New York, NY, USA, 1999, p. 7–14.

P2P

Systèmes répartis de grande taille : – de l'anarchie à l'auto-structuration

Gilles Trédan[1] **et Aline C. Viana**[3]

[1] *Université de Rennes 1-IRISA, France*
[3] *INRIA Saclay - Ile de France sud, France*

RÉSUMÉ. *L'auto-structuration est une propriété très difficile à établir dans les réseaux de grande taille lorsque les entités n'ont accès à aucune information globale concernant le réseau qu'elles composent. Dans ce contexte, il est crucial de permettre aux entités de se positionner les unes par rapport aux autres. Ceci peut-être obtenu en définissant un système de coordonnées auquel les entités pourront se référer pour obtenir leur position. Un tel système fournit alors une brique de base pouvant être utilisée pour l'auto-structuration. Dans cet article, nous présentons un tel système de coordonnées : VINCOS. Ce système, léger et complètement décentralisé, repose uniquement sur l'exploitation de données locales et la communication entre entités voisines. C'est une grande différence par rapport à la plupart des approches existantes qui reposent soit sur des référentiels prédéfinis, soit sur des systèmes de positionnement auxiliaires. Des simulations montrent ensuite que cette approche est à la fois efficace et précise.*

ABSTRACT. *Self-structuring is extremely hard to achieve when no global information about the network is available. Yet, it is crucial in such contexts, that each node be able to infer its position with respect to the other nodes. One solution for this is to allow the nodes to access a coordinate system from which they can obtain a coordinate assignment. Such a coordinate system represents the basic layer on top of which adaptive behaviors can be designed. In this paper, we present the design and the evaluation of a simple lightweight and fully decentralized virtual coordinated system (VINCOS), which relies only on local connectivity information and per-neighbor communication. This is to oppose to most existing approaches relying either on pre-defined positioning referential or signal measurement. We also show how the VINCOS coordinates can be used to enable a large set of nodes to configure themselves in arbitrary geometric structures. The evaluation demonstrates that the approach is both efficient and accurate.*

MOTS-CLÉS : *Système autonomes, structuration géométrique, réseau auto-organisant, réseaux de capteurs, système de coordonnées virtuelles*

KEYWORDS: *Autonomous system, Geometric structuring, Locality, Self-organizing network, Sensor network, Virtual coordinate system*

1. Introduction

L'auto-structuration représente la capacité d'un système à créer un structure spécifique, en partant de rien, et sans apport d'information extérieure. C'est une dimension importante des systèmes autonomes, par exemple pour le passage à l'échelle [DOH 07]. La complexité d'un telle structuration dépend de la quantité d'information dont les entités (ou noeuds) disposent initialement. Nous définissons par *connaissance externe* l'information procurée aux noeuds par une entité ou un dispositif externe à celui-ci. A l'opposée, la *connaissance intrinsèque* est la connaissance que les noeuds établissent eux-mêmes par l'observation du réseau. D'un coté, l'utilisation de connaissance externe réduit l'autonomie du réseau, puisque son fonctionnement dépend alors d'une hypothèse supplémentaire : la disponibilité de cette connaissance. De l'autre, se baser sur de la connaissance intrinsèque renforce fortement l'autonomie du système, mais au prix de plus de communication. Ceci peu réduire considérablement l'espérance de vie du système. En résumé, l'autonomie d'un système est inversement proportionnelle à la quantité de connaissances extérieures qu'il faut lui apporter pour lui permettre de se structurer. Il est néanmoins crucial d'établir un bon compromis entre autonomie et coûts de communications.

Cet article propose un système adaptatif de coordonnées virtuelles pour les réseaux de capteurs (ou *VINCOS* pour VIrtual Networked COordinate System). Ce système constitue une brique de base pour la structuration du réseau, et compose ainsi le coeur d'un système autonome. La grande différence avec les travaux existants est qu'il ne repose sur aucune ancre, ne suppose aucun noeud conscient de sa position, et n'utilise aucun système de mesure du signal. Les noeuds obtiennent des coordonnées d'une manière complètement décentralisée. Pour cela ils exploitent uniquement des informations locales ou obtenues par communications entre noeuds voisins.

La suite de cet article est organisée ainsi : après avoir introduit notre modèle dans la section 2, nous présentons une étude des travaux semblables en section 3. VINCOS est décrit section 4. Sa performance est évaluée en section 6. Enfin, la section 7 conclut en présentant les perspectives de recherche.

2. Modèle

Nous nous concentrons sur les scénarios où la région surveillée ne permet pas d'intervention humaine. Pour des raisons de simplicité, l'analyse des coûts sera effectuée soit sur une distribution de noeuds uniforme dans une zone de forme rectangulaire, soit en supposant les noeuds répartis régulièrement sur une grille. Le principe demeure le même sur d'autres distributions. Dans le cas d'une grille de N noeuds, chaque rangée et colonne contient $\sqrt{N}$ noeuds.

Noeuds Le système est composé d'un ensemble fini de N capteurs répartis de façon uniforme sur une zone géographique[1]. Chaque noeud possède un identifiant unique i.

1. Une fois de plus, pour des raisons de simplicité, la distribution est supposée uniforme. Bien qu'il soit trop tôt pour affirmer que notre algorithme converge indépendamment de la distribution, les simulations que nous avons effectué sur des distributions plus exotiques vont dans ce sens

Communication Chaque noeud peut communiquer avec tout autre noeud situé à portée radio, qui est modélisée par un disque de diamètre R. On suppose le réseau non partitionné et les communications bidirectionnelles.

Connaissance initiale Initialement, un noeud connait seulement son identité, le fait qu'il soit le seul à posséder cette identité dans le réseau, et un paramètre, d, définissant la dimension de l'espace de coordonnées.

3. Contexte

La communauté des réseaux sans fils propose une vaste littérature concernant les systèmes de positionnement [HOL 01, glo, BEN 06, BEN 07, CAP 01, BEN 05, CAR 05, BIS 04, SHA 03, NIC 03, GUS 03, DOH 01, RAO 03, MOS 04]. La majorité de ces approches reposent sur des hypothèses spécifiques (de la disponibilité d'informations GPS à la connaissance de l'angle d'arrivée du message radio), et témoignent de la nécessité d'une approche permettant plus d'autonomie. Cela fut traité dans certaines approches n'utilisant aucune *ancre* (noeud connaissant sa position). Il en résulte des *systèmes de coordonnées virtuelles* [RAO 03, MOS 04, BEN 06] (par opposition aux systèmes de coordonnées géographiques).

Ces travaux présentent clairement les enjeux et les solutions pour l'établissement de coordonnées virtuelles. Ces solutions sont cependant très coûteuses, tant en temps de calcul qu'en nombre de messages, et peu applicables aux réseaux de capteurs [RAO 03, MOS 04]. De plus, certaines de ces méthodes, bien que n'ayant pas recours aux ancres, requièrent l'utilisation de noeuds *bien placés* dans le système afin de servir comme noeuds de référence [MOS 04, BEN 06, BEN 07].

4. VINCOS : un système de coordonnées virtuelles économique

Les techniques présentées ici reposent sur des principes simples. Encore une fois, cette approche n'utilise que les capacités naturelles et intrinsèques des communications radio (connexité, nombre de messages reçus et observations de voisinage) pour *1)* identifier les zones denses du système, et *2)* utiliser ces zones denses pour détecter la périphérie (ou *bordure*) du système. Ces information peuvent ensuite être exploitées pour implémenter différentes fonctionnalités.

4.1. *Principe des coordonnées*

Le positionnement est une information clé dans la construction et le maintien de systèmes autonomes. Un système de coordonnées procure à chaque noeud une *position* qui est à la fois localement et globalement cohérente. Dans VINCOS, chaque noeud est configuré initialement avec un paramètre d : la dimension du système de coordonnées. Ainsi, les coordonnées virtuelles d'un noeud i sont représentées sous la forme d'un tuple $(x_1, \ldots, x_d)$, où x_j est la projection du noeud i sur le $j$ème axe de l'espace virtuel à d dimensions. L'unité de distance utilisée est le saut.

Cet espace virtuel est définit comme suit. La bordure de la zone géographique couverte par les noeuds est partitionnée en d "segments". Ces *segments de bordure* peuvent avoir soit la même taille, soit des tailles différentes[2]. Soit un axe $j, 1 \leq j \leq d$, du système de coordonnées. La coordonnée x_j est alors la distance, en sauts, du plus court chemin du noeud i au segment j (*i.e.* au noeud le plus proche appartenant au segment). La figure 1 illustre ceci : l'espace de coordonnées y est de dimension $d = 3$. Les coordonnées virtuelles de trois noeuds y sont portées : les coordonnées $(2, 4, 2)$ signifient que le noeud est à distance 2 des bordures 1 et 3, et à distance 4 de la bordure 2. Ce système ne nécessite ainsi pas d'ancre ou de noeud particulier.

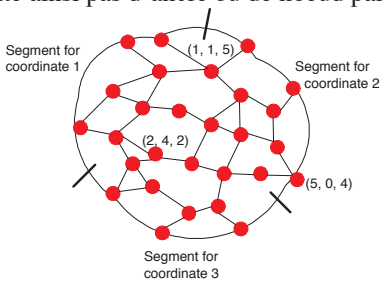


Figure 1. *Un exemple de coordonnées virtuelles.*

Les coordonnées de VINCOS dépendent d'une définition précise des segments de bordure, *i.e.* les d segments qui composent la bordure du système. Afin de définir de tels segments, VINCOS repose sur la construction d'une *ceinture* de processus autour du système qui épouse la bordure (les noeuds la bordure). Détecter quels sont les noeuds appartenant à la bordure de manière précise est difficile. Nous allons voir comment VINCOS calcule une telle *ceinture*.

4.2. *VINCOS : De l'anarchie aux coordonnées virtuelles*

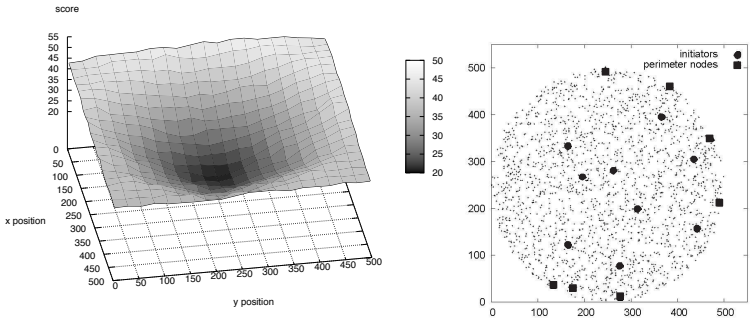
Afin de construire un système de coordonnées cohérent de façon décentralisée, VINCOS comporte quatre phases. Nous allons ici les détailler.

4.2.1. *Phase 1 : detection des initiateurs*

La première phase utilise les densités pour identifier un petit ensemble d'*initiateurs*. Ce sont des noeuds qui joueront un rôle important dans les phases suivantes.

Découverte des voisins Chaque noeud i émet un message contenant son identité. Ainsi chaque noeud i peut établir la liste des noeuds dans son voisinage : l'ensemble $neighbors_i$. Chaque noeud i émet ensuite son nombre de voisins, $|neighbors_i|$. Cela permet à chaque noeud d'apprendre le nombre de voisins qu'ont ses voisins.

2. Pour des raisons de simplicité, nous présenterons le cas où les segments sont de taille égale. La démarche est identique si les segments ont des tailles différentes



(a) (b)

Figure 2. *(a) Les noeuds de bordure ont un score plus élevé.(b) Initiateurs et initiateurs de bordure pour un réseau de* 2000*-noeuds.*

Découverte des initiateurs Un initiateur est définit comme un maxima local de densité. Ceci se traduit intuitivement par le noeud ayant plus de voisins que tous ses voisins. Formellement, un noeud i est initiateur s'il vérifie le prédicat suivant :

$$\forall j \in neighbors_i : \quad |neighbors_i| > |neighbors_j|. \qquad [1]$$

Puisque l'on suppose une densité uniforme, les noeuds sont supposés avoir un nombre de voisins proportionnel à l'aire du système présent à portée radio. Ainsi les noeuds de bordure, qui ont la moitié de leur disque radio hors du système, sont supposés avoir moitié moins de voisins que les noeuds dans le système. Ainsi, le prédicat 1 permet de sélectionner des voisins qui ne sont pas sur la bordure, ce qui a son importance pour les phases suivantes. Des simulations sur un grand nombre de scénarios ont montré que ce nombre reste bas (souvent 3,4 ou 5).

Complexité Le coût total de cette phase est de $2N$ messages, et la convergence est obtenue en 2 étapes.

4.2.2. *Phase 2 : définition du score de bordure*

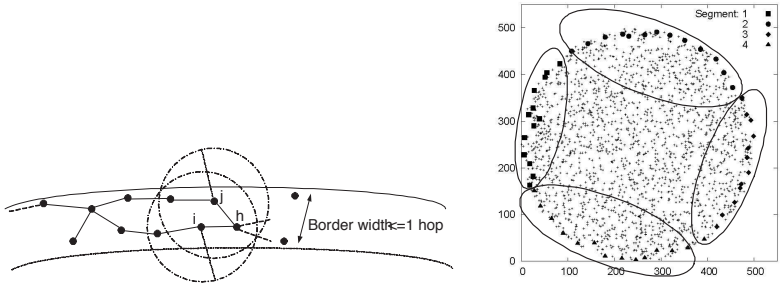
L'objectif est de procurer à chaque noeud un *score de bordure*, définit ensuite. Pour cela, chaque noeud détermine sa distance à chaque initiateur.

Calcul de la distance aux initiateurs Chaque initiateur i émet un message m contenant son identité et un compteur de sauts, initialisé à 0. Après avoir reçu le message m, et si m améliore sa distance à i, un noeud j incrémente le compteur de sauts de m et le réémet ; sinon il ne tient pas compte du message m. La distance d'un noeud j à un initiateur i est : $dist(i,j) = min\{dist(i,\ell)|\ell \in neighbors_j\} + 1$.

Le score de bordure La valeur, notée $score_j$, du *score de bordure* d'un noeud j est définie ainsi :

$$score_j = \sum_{i \in initiators} dist(i,j). \qquad [2]$$

Il est intéressant d'observer qu'en plus de connaître son score, chaque noeud connait aussi le score de ses voisins (puisque chaque noeud réémet les messages provenant des initiateurs). Ce score peut être vu comme la distance à un "initiateur moyen". La figure 2(a) représente, en échelle de gris, le score de chaque noeud en fonction de sa position. On peut constater que le minimum (*i.e.* le plus proche de l'initiateur moyen) est situé au centre du système.



(a) (b)

Figure 3. *(a) Les voisinages des noeuds i et j s'intersectent, ce qui permet au noeud h de ne pas croire qu'il a reçu des sondes provenant de côtés différents. (b) Les 4 segments définis pour un réseau de 2000 noeuds.*

Quelques noeuds de la bordure Le deuxième objectif de la phase 2 est de découvrir quelques noeuds qui sont *de facon certaine* sur la bordure. Nous les appellerons *initiateurs de bordure*. Ces noeuds sont les noeuds responsables de l'initiation de la construction de la ceinture, lors de la phase 3. Un noeud j se découvre en bordure par rapport à un initiateur i s'il vérifie le prédicat suivant :

$$\forall \ell \in neighbors_j : \quad dist(i,j) \geq dist(i,\ell). \qquad [3]$$

Soit x un pourcentage d'initiateurs. Le protocole considère qu'un noeud est initiateur de bordure s'il est en bordure par rapport à $x\%$ d'initiateurs. Autrement dit, un noeud est initiateur de bordure s'il vérifie le prédicat 3 pour $x\%$ initiateurs. Nous avons expérimentalement constanté que $x = 60\%$ permettait un bon compromis entre faux positifs (noeuds détectés à tort comme en bordure) et faux négatifs (noeuds non détectés comme en bordure). La figure 2(b) illustre un exemple de sélection d'intiateurs et d'initiateurs de bordure.

Complexité Pour N noeuds et y initiateurs, cette phase a un coût en communication de $O(y * N)$ messages. Si l'on considère les noeuds répartis régulièrement sur une

grille, le temps de cette phase est proportionnel à la distance maximale entre deux noeuds du réseau, soit $\sqrt{2N}$ (la diagonale de la grille).

4.2.3. *Phase 3 : Construction de la ceinture*

Cette phase est le coeur de VINCOS. L'objectif est d'établir une liste de noeuds qui appartiennent tous à la bordure. Cette liste est une structure mono-dimensionnelle, ce qui facilitera son exploitation ultérieure. La détection d'une telle bordure est complexe s'il faut à la fois un coût raisonnable et peu de faux positifs/négatifs.

Lancement des sondes Chaque initiateur de bordure identifié à la phase précédente envoie un message, appellé *sonde*, qui va "coller" à la bordure du système en s'appuyant sur les scores de bordure.

Transmission de la sonde Une sonde contient deux ensembles de noeuds : $destinators$ et $exclude$. Un noeud i considère qu'il reçoit une sonde ssi l'ensemble $destinators$ de la sonde contient i. Rappelons que chaque noeud connait son score de bordure, et le score de tous ses voisins. Lorsque le noeud i reçoit une sonde, il sélectionne k candidats à qui il va relayer la sonde (k est un paramètre de l'application). Ces k noeuds sont les noeuds ayant **le score de bordure le plus élevé** parmi l'ensemble $candidates_i$ tel que : $candidates_i = neighbors_i - (destinators \cup exclude)$.

Ces k noeuds constituent l'ensemble $destinators$ de la sonde que i va émettre. L'ensemble $exclude$ de la sonde émise par i est l'ensemble $destinators$ de la sonde que i a reçu. La sonde n'étant relayée que par les noeuds ayant un score de bordure localement maximal, elle va "coller" la bordure.

Chaque sonde contient aussi un champ $path$, qui contient la liste des noeuds qui ont relayé la sonde : chaque noeud relayant la sonde y ajoute son identité. Enfin, chaque sonde contient aussi la liste des voisins du deuxième noeud la relayant. Cette liste est appelée $stamp$. Elle sert à identifier de manière unique chaque "coté" de la ceinture : deux sondes pourraient emprunter le même coté de la bordure avec des chemins différents (voir Figure 3(a)), mais l'intersection de leurs champs $stamp$ serait alors non nulle.

Définition des segments Lorsqu'un noeud j reçoit deux sondes issues du même initiateur de bordure, il vérifie que ces deux sondes ont des chemins et des champs $stamp$ différents. Si c'est le cas, l'union des champs $path$ des deux sondes est une ceinture du système. Le noeud j, alors appelé *définisseur de segment*, va découper cette ceinture en d segments, et émettre un *message de segment* contenant la liste des noeuds composant chaque segment. Ce message de segment va alors emprunter le chemin inverse, relayé par les noeuds de la ceinture. Chaque noeud relayant un tel message apprend alors à quel segment il appartient (voir figure 3(b)).

Combattre l'ambiguïté Un mécanisme de propagation-extinction est utilisé sur les identités des initiateurs de bordures afin de ne garder qu'une segmentation pour tout les système.

Complexité Le temps nécessaire est proportionnel au nombre de noeuds de la ceinture, soit $4\sqrt{N}$ dans le cas de la grille. Le coût en message est de $z * 4\sqrt{N}$, où z est le nombre d'initiateurs de bordure.

4.3. *Phase 4 : Définission des coordonnées*

Obtention des coordonnées Lorsqu'un noeud de la ceinture a appris à quel segment s auquel il appartient, il émet un message contenant ce numéro s et un compteur de sauts initialisé à 0. Ces messages sont relayés et utilisés par tous les noeuds pour découvrir à quelle distance ils sont de chaque segment de bordure. Chaque noeud i possède finalement un enssemble de coordonnées $(x_1, \ldots, x_d)$ dans l'espace de coordonnées d-dimensionnel.

Complexité Après définition des segments, d types de messages différents se propagent à travers le réseau. Chaque noeud ne relaie qu'un message par numéro de segment (celui qui a le plus petit compteur de sauts). Le coût en messages de cette phase est donc de $d \times N$. Les d messages se propagent simultanément, le temps de convergence est donc de $\sqrt{2N}$ étapes.

5. Des coordonnées virtuelles à la structuration géométrique du réseau

Les coordonnées sont principalement utilisées pour faire du routage. Elles présentent selon nous bien d'autres intérêts. Elles constituent entre autres un outil puissant d'organisation du réseau. Une contribution de cet article consiste à présenter comment les exploiter à de telles fins.

Afin d'organiser un réseau, il est intéressant de pouvoir assigner aux noeuds différents comportements selon leur position. Bien que ce soit possible en utilisant n'importe quel système de coordonnées, nous montrons ici qu'il est possible de le faire en utilisant les coordonnées assignées par VINCOS. L'objectif ici est d'utiliser les coordonnées VINCOS afin de faire émerger une structuration géométrique du réseau. Il sera ensuite possible d'utiliser cette structuration à de multiples fins : clustering, aggrégation de données, gestion des consommations d'énergie. Une fonction de structuration géométrique est une fonction transformant des coordonnées en un numéro de partition. Formellement, soit $\mathcal{K}$ l'espace des coordonnées (avec VINCOS, $\mathcal{K} = \mathbb{N}^d$), et p le nombre de partitions différentes souhaitées, alors une fontion de structuration géométrique est une fonction f telle que :

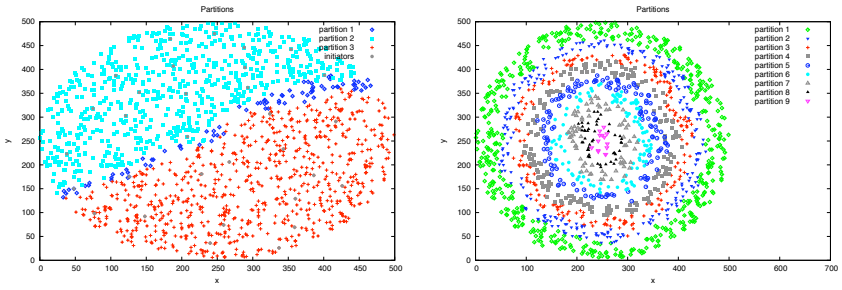
$$\begin{array}{rcl} f : \mathcal{K} & \rightarrow & \{0, \ldots, p\} \\ f(c_i) & \rightarrow & p_i. \end{array}$$

Une telle définition permet à un noeud de calculer le numéro de partition de n'importe quel noeud dont il connait les coordonnées : *chaque noeud a une idée de la structuration du système*.

Prenons un example : un ensemble de capteur disséminés sur une grande zone. On assigne trois groupes à ces capteurs : *Nord*, *Sud* et *Equateur*. Le groupe Equateur est une “ligne droite de noeuds” qui marque la séparation entre les groupes Nord et Sud. Considérons une application où les capteurs doivent collecter et rassembler de l'information (via un protocole de routage). Les capteurs envoient l'information au groupe Equateur. Ainsi, périodiquement, un avion survole la ligne Equateur pour collecter toute l'information. Obtenir un tel partitionnement en trois groupes est facile. Soit (x_1, x_2) les coordonnées VINCOS ($d = 2$) d'un noeud i. La fonction de structuration est alors (voir figure 4(a)) :

$$\begin{array}{rcl} f : \mathbb{N} * \mathbb{N} & \rightarrow & \{1, 2, 3\} = \{South, Equator, North\} \\ f(x_1, x_2) & \rightarrow & \left\{ \begin{array}{lll} 1 & when & x_1 > x_2 \\ 2 & when & x_1 = x_2 \\ 3 & when & x_1 < x_2 \end{array} \right. \end{array}$$

Un autre partionnement géographique est “la cible” (voir figure 4(b)). C'est une structuration facile à obtenir avec VINCOS et ne nécessitant qu'un segment ($d = 1$). La fonction f permettant de l'obtenir est alors l'identité : chaque noeud a pour numéro de partition sa distance en sauts à la bordure la plus proche.



(a) (b)

Figure 4. *(a)-(b) Exemples de structuration géométrique.*

6. Evaluation des performances

Cette section décrit les expériences que nous avons effectué pour attester l'efficacité et la précision de nos propositions. Les expériences ont été conduites avec un simulateur discret écrit en Java. Nous sommes intéressés par l'évaluation algorithmique du protocole, aussi le simulateur n'implémente pas une couche Mac et un modèle de communication réaliste, mais une version simplifiée sans perte de messages, ni collisions[3].

3. Nous pensons que les pertes et les collisions, bien qu'impactant le temps nécessaire, n'empêcheraient pas la convergence du protocole. Etudier leur impact en détail constitue une direction de recherche

Nous simulons des réseaux comportant 250 à 2600 noeuds. Ces noeuds sont distribués sur une zone $2d$ de 500×500 unités de distance. La zone de portée radio est un disque dont le diamètre varie de 30 à 50 unités de distance.

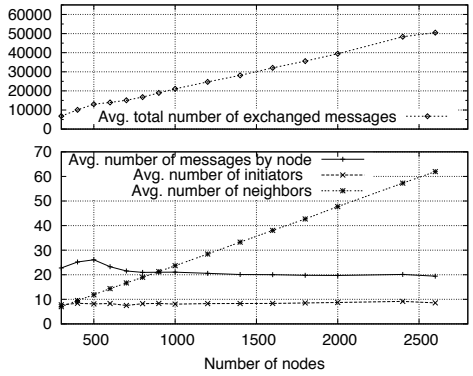
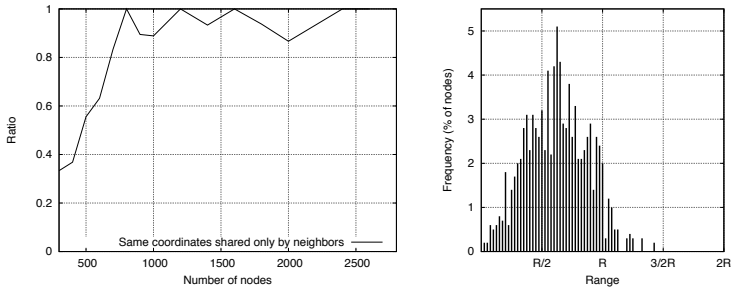


Figure 5. *Analyse du coût en fonction de la taille du réseau, pour* $d = 4$*, portée radio de 40-unités.*

Coûts de communication Le coût total de communication de VINCOS est $O((2 + y + d) * N + 4 * z\sqrt{N})$, avec N le nombre total de noeuds, y et z les nombres d'initiateurs et d'initiateurs de bordure, resp., et d le nombre de dimensions du système de coordonnées.



(a) (b)

Figure 6. *(a) Ratio des simulations ayant conduit à des systèmes précis. (b) Histograme des distances entre noeuds homonymes dans un système non précis de* 1000 *noeuds.*

La figure 5 confirme le résultat théorique : le coût d'exécution de VINCOS par noeud est bas, et indépendant de la taille du réseau. On peut observer que malgré les variations de taille du réseau et de densité, le nombre d'initiateurs reste stable et bas. De plus, même si le nombre total de message augmente avec la taille du réseau, le

coût par noeud décroît légèrement à mesure que le système croît. Ceci est dû au fait que le nombre de messages échangés par les noeuds internes reste constant et faible, mais que leur proportion dans le système augmente. Ainsi plus le système comporte de noeuds, plus la proportion de noeuds de bordure baisse au profit des noeuds internes. Or ce sont les noeuds de bordure qui génèrent la majorité des messages échangés. Donc le coût moyen par noeud baisse[4].

Précision Bien que le routage ne soit pas l'objet de cet article, il est intéressant d'observer la précision de VINCOS. Il est reconnu [BEN 06] qu'un système de coordonnées précis permet un routage glouton efficace. Afin d'observer cette précision, et par conséquent les capacités de routage de VINCOS, nous avons étudié la répartition des noeuds partageant les mêmes coordonnées.

Nous appellerons *système précis* un système vérifiant l'assertion : "si deux noeuds partagent les mêmes coordonnées alors ils sont voisins". Notons qu'un système précis permet un bon routage glouton, puisqu'un message arrivant aux coordonnées finales est arrivé à destination.

Nous avons effectué 20 expériences indépendantes pour différentes tailles de système, et une portée radio de 50 unités de distance. La figure 6(a) montre, en fonction de la taille du système simulé, quel est le ratio des expériences ayant memé à un système précis. On peut observer que, pour certaines tailles, tous les systèmes simulés étaient précis. Pour une taille de 1000 noeuds, 90% des systèmes sont précis.

La figure 6(b) montre le détail d'un système de 1000 noeuds qui n'est pas précis. Afin de montrer qu'une grosse partie du système est tout de même précise, cette figure représente la répartition des distances physiques entre des noeuds partageant les mêmes coordonnées (des *homonymes*). R est la portée radio (ici, $R = 50$), donc les homonymes distants de plus de R unités ne sont pas voisins. Ainsi, on peut y lire que $3,2\%$ des homonymes sont distants de $R/2$ unités. On observe que la majeure partie des homonymes sont voisins, et qu'il n'y a pas d'homonymes distants de plus de $3/2 \times R$. On peut conclure que, même si tous les homonymes ne sont pas voisins, la majeure partie le sont, et qu'un routage glouton dans un tel système ne poserait pas de problèmes.

7. Conclusion

Cet article a présenté un algorithme, *VINCOS*, qui assigne des coordonnées virtuelles aux noeuds, et montré que l'exploitation de ces coordonnées permet de structurer un réseau. VINCOS repose sur des hypothèses très faibles (les noeuds ont des identités distinctes et comparables), et ne nécessiste aucune ancre ou système de mesure de signal. Il est de plus très économique et complètement décentralisé. La contribution clé dans VINCOS est l'algorithme de détection de bordures. C'est lui qui assure la construction économique d'un système de coordonnées fiables. Les simulations attestent de la précision de ce système. Il serait intéressant d'étudier le cas de systèmes

4. Ce n'est pas le cas dans [RAO 03], puisque chaque noeud de bordure inonde le réseau

aux frontières plus singulières, et d'explorer de manière plus approfondie les possibilités de la structuration géométrique pour des applications particulières.

8. Bibliographie

[BEN 05] BENBADIS F., FRIEDMAN T., AMORIM M. D., FDIDA S., « GPS-free-free positioning system for sensor networks », *Proc. of WOCN*, avril 2005.

[BEN 06] BENBADIS F., PUIG J.-J., AMORIM M. D., CHAUDET C., FRIEDMAN T., SIMPLOT-RYL D., « JUMP : Enhancing hop-count positioning in sensor network using multiple coordinates », rapport, avril 2006, TR arXiv cs.NI/0604105.

[BEN 07] BENBADIS F., OBRACZKA K., CORTÈS J., BRANDWAJN A., « Exploring landmark placement strategies for self-organization in wireless sensor networks », *Proc. of IEEE PIMRC*, septembre 2007.

[BIS 04] BISCHOFF R., WATTENHOFER R., « Analyzing connectivity-based, multi-hop ad hoc positioning », *Proc. of IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, mars 2004.

[CAP 01] CAPKUN S., HAMDI M., HUBAUX J.-P., « GPS-free positioning in mobile ad hoc networks », *Proc. of the 34th Annual Hawaii International Conference on System Sciences*, janvier 2001.

[CAR 05] CARUSO A., CHESSA S., DE S., URPI A., « GPS free coordinate assignment and routing in wireless sensor netwoks », *Proc. of IEEE Infocom*, mars 2005, p. 150–160.

[DOH 01] DOHERTY L., PISTER K. S. J., GHAOUI L. E., « Convex position estimation in wireless sensor networks », 2001, p. 1655–1663.

[DOH 07] DOHLER M., WATTEYNE T., BARTHEL D., VALOIS F., LU J.-L., « Kumar's, Zipf's and Other Laws : How to Structure an Optimum Large-Scale Wireless (Sensor) Network ? », *13th European Wireless Conference*, Paris, France, avril 2007.

[DOL 00] DOLEV S., « *Self-stabilization* », The MIT Press, 2000.

[glo] « The Glonass positioning system », `www.glonass-center.ru/`.

[GUS 03] GUSTAFSSON F., GUNNARSSON F., « Positioning using time-difference of arrival measurements », *Proceedings of ICASSP*, 2003.

[HOL 01] HOLFMANN-WELLNHOF B., LICHTENEGGER H., COLLINS J., « *Global Positioning Systems : Theory and Practice* », Springer, 5th edition, 2001.

[MOS 04] MOSCIBRODA T., O'DELL R., WATTENHOFER M., WATTENHOFER R., « Virtual Coordinates for Ad hoc and Sensor Networks », *Proc. of Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, octobre 2004.

[NIC 03] NICULESCU D., NATH B., « Ad Hoc Positioning System APS using AoA », *Proc. of IEEE Infocom*, mars 2003.

[PEA 80] PEASE L., SHOSTAK R., LAMPORT L., « Reaching Agreement in Presence of Faults », *Journal of the ACM*, vol. 27, n° 2, 1980, p. 228–234.

[RAO 03] RAO A., RATNASAMY S., PAPADIMITRIOU C., SHENKER S., STOICA I., « Geographic Routing without Location Information », *Proc. of ACM Mobicom*, avril 2003.

[SHA 03] SHANG Y., RUML W., ZHANG Y., FROMHERZ M., « Localization from mere connectivity », *Proc. of ACM Mobihoc*, juin 2003.

Caractérisation du trafic P2P IPTV [1] : une analyse d'échelle

Thomas Silverston* — **Olivier Fourmaux*** — **Kavé Salamatian****

** UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France*
{thomas.silverston, olivier.fourmaux}@lip6.fr

*** Lancaster University, InfoLab21, LA1 4WA, United Kingdom*
{kave.salamatian}@lancs.ac.uk

RÉSUMÉ. *Les applications de diffusion de flots multimédia en direct via un réseau pair à pair (P2P IPTV) sont en plein essor et promises à un bel avenir. Leur utilisation massive va fortement augmenter le trafic réseau de l'Internet. C'est pourquoi il est important de caractériser leur trafic afin d'évaluer leurs effets sur le réseau. Dans cet article, nous étudions le trafic d'applications P2P IPTV. Nous avons pour cela mesuré leur trafic et nous caractérisons sa structure à différentes échelles temporelles en utilisant une méthode de transformées en ondelettes. Notre analyse montre que les propriétés d'échelle du trafic P2P IPTV sont différentes suivant que les applications utilisent principalement TCP ou UDP. Les caractéristiques du trafic descendant sont différentes de celles du trafic montant. La signalisation utilisée a également un impact significatif sur les propriétés du trafic descendant, mais n'affecte pas le trafic montant. Ces observations nous permettront de concevoir des modèles de trafic P2P IPTV réalistes qui sont des éléments essentiels pour modéliser ou simuler correctement ces systèmes.*

ABSTRACT. *P2P IPTV applications arise on the Internet and will be massively used in the future. It is expected P2P IPTV will contribute to increase the overall Internet traffic. In this context, it is important to measure their traffic to evaluate the impact on the network and characterize this traffic. In this paper, we measured the network traffic of P2P IPTV applications. From the collected data, we characterize the P2P IPTV traffic structure at different time scales by using wavelet based transform method. Our analysis shows that the scaling properties of the TCP traffic is different from the UDP traffic. The download traffic has different characteristics than the upload traffic. The signaling traffic has also a significant impact on the download traffic but it has negligible impact on the upload traffic. These findings will help to design synthetic P2P IPTV traffic generation models, which are key input parameters when modeling or simulating these systems.*

MOTS-CLÉS : *mesures de trafic, analyse de trafic, pair à pair, vidéo en direct*

KEYWORDS: *traffic measurement, traffic analysis, peer to peer, live video*

1. Télévision diffusée sur Internet via un réseau pair à pair

1. Introduction

Les applications pair à pair de diffusion continue en direct (P2P live streaming) comme les applications de diffusion de télévision sur Internet via un réseau P2P (P2P IPTV) se répandent sur Internet et seront largement utilisées dans le futur. Actuellement le trafic P2P, du fait des applications P2P de partage de fichiers telles que BitTorrent [COH 03] ou eDonkey [EDO], compte déjà pour une part importante du trafic de l'Internet. L'apparition de nouveaux services de diffusion de vidéo comme Youtube [YOU] contribue également à l'augmentation du trafic de l'Internet. L'intégration de la technologie P2P avec de nouveaux services vidéo va décupler encore le trafic d'Internet. Ainsi, il est important d'étudier l'impact de ce trafic sur le réseau et d'en caractériser les propriétés.

Il y a deux raisons principales pour étudier le trafic P2P IPTV : tout d'abord, cela permet de comprendre les propriétés de ce trafic et leur impact sur le réseau. Ensuite, grâce à ces observations, il sera alors possible de concevoir des modèles synthétiques de trafic d'applications P2P IPTV. C'est d'ailleurs une motivation essentielle puisque la plupart des protocoles proposés [VEN 06] [MAG 07] sont validés en utilisant des outils de simulation ou bien des modèles mathématiques qui n'utilisent pas des modèles de trafic réalistes comme paramètres d'entrée de leurs études. Le trafic P2P IPTV présente des contraintes de qualité de service (QdS) importantes (bande passante, délais, gigue) comparé à d'autres trafics P2P (transfert de fichiers P2P) et l'utilisation de modèles de trafic inappropriés biaiserait les résultats des simulations et modélisations.

Il existe déjà des études du trafic P2P IPTV [HEI 07][ALI 06][VU 07][SIL 07a], mais elles se focalisent sur l'ingénierie inverse d'applications commerciales. Dans ce cas précis, les mesures de trafic et leurs analyses sont les seules façons de déduire les mécanismes de ces applications. Notre étude est différente puisque nous cherchons à caractériser les propriétés globales du trafic P2P IPTV. Dans cet article, nous présentons une analyse multi échelle de la structure du trafic P2P IPTV. Pour cela, nous avons collecté le trafic des applications P2P IPTV les plus populaires. Le comportement multi échelle du trafic est analysé en utilisant un outil basé sur les transformées en ondelettes. Ces travaux sont, à notre connaissance, les premiers à caractériser le trafic P2P IPTV à différentes échelles temporelles et à présenter ses propriétés.

Notre analyse de trafic révèle des différences significatives dans le comportement d'échelle du trafic suivant que l'application utilise principalement TCP ou UDP. Le trafic TCP présente des comportements périodiques alors que le trafic UDP présente des caractéristiques de dépendance à long terme, ce qui affectera la qualité de la réception vidéo. Le trafic de signalisation a quant à lui un impact sur le trafic descendant (*download*) mais n'a pas d'impact sur le trafic montant (*upload*). Le trafic descendant a des propriétés d'échelles différentes du trafic montant et chacune des directions du trafic doit être prise en compte pour concevoir judicieusement des modèles de trafic P2P IPTV.

Dans cet article, nous présentons d'abord l'état de l'art dans la section 2. Puis dans la section 3, nous décrivons notre plateforme expérimentale de mesure ainsi que les

applications utilisées. Nous définissons notre méthodologie pour analyser le trafic à différentes échelles temporelles dans la section 4 et nous présentons les résultats de cette analyse multi échelle du trafic P2P IPTV dans la section 5. Enfin, nous concluons l'article à la section 6 et exposons nos futures perspectives de recherche.

2. Etat de l'art

Actuellement, les applications P2P IPTV sont de plus en plus étudiées via l'analyse de leur trafic. Ainsi, [ZHA 05a] présente les premiers résultats de mesures du protocole Donet [ZHA 05b], qui fut déployé sur l'Internet (Coolstreaming). Cette étude fournit des statistiques telles que le comportement des utilisateurs dans de tels systèmes P2P ainsi que des indications sur la qualité de la vidéo reçue par les utilisateurs. PPLive, La plus populaire des applications P2P IPTV, a été largement analysée [HEI 07]. Les auteurs ont effectué des mesures actives du réseau P2P en instrumentalisant leur propre client PPLive. Ils fournissent ainsi de nombreux détails d'implémentation comme la taille des mémoires tampons utilisées ou encore le nombre de pairs dans le réseau P2P. D'autres mesures actives de PPLive ont été réalisées [VU 07] et obtiennent des modèles mathématiques concernant la distribution des utilisateurs par chaînes retransmises ou bien la durée des sessions vidéos.

Dans nos précédents travaux [SIL 07a], nous avons mesuré de façon passive le trafic réseau généré par les applications les plus populaires sur l'Internet durant un événement mondial. Nos mesures nous ont permis de déduire certains mécanismes utilisés par ces applications. Nous avons ainsi comparé les applications entre elles afin de mettre en évidence leurs similarités et leurs différences. Les mécanismes de SOPCast et PPLive ont également été analysés par des mesures passives de leur trafic [ALI 06].

En résumé, tous ces travaux étudient les systèmes P2P IPTV en analysant leur trafic, mais il n'existe encore aucune étude cherchant à caractériser la structure de corrélation du trafic généré à différentes échelles temporelles afin de comprendre les propriétés de ce trafic et leurs effets sur le réseau.

3. Expériences de mesure

Afin de mesurer le trafic P2P IPTV pendant un évènement présentant de l'intérêt à être regardé en direct, nous avons collecté ce trafic durant la coupe du monde de football 2006. Nous avons mesuré la plupart des matchs de football avec différentes applications en même temps et récolté une importante quantité de données (> 250 Go). Nous avons choisi d'utiliser plusieurs applications pour être capable de caractériser les propriétés du trafic P2P IPTV sans être dépendant d'une seule implémentation. Les applications mesurées seront présentées dans la partie 3.2. Devant le volume important de données récoltées, nous nous intéresserons dans cet article à des expériences de mesures effectuées le 30 juin 2006 alors que deux matchs étaient diffusés. L'analyse de traces de trafic d'autres matchs nous a montré que celles considérées dans cet article sont représentatives de tout notre jeu de données.

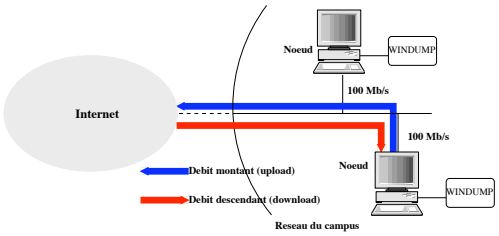


Figure 1. *Plateforme expérimentale de mesure. Chaque noeud est un PC standard directement connecté à l'Internet à travers le réseau du campus universitaire.*

3.1. *Plateforme expérimentale de mesure*

Notre plateforme expérimentale de mesures est décrite sur la figure 1. Pour collecter le trafic, nous avons utilisé deux ordinateurs équipés d'un processeur à 1.8GHz et disposant de cartes graphiques aux capacités standard. Le système d'exploitation utilisé sur les ordinateurs était Windows XP puisque les applications mesurées ont seulement été implémentées pour celui-ci. Les applications utilisent des codecs vidéo de type MPEG4. Les ordinateurs (noeuds) étaient situés dans le réseau de notre campus universitaire et étaient directement connectés à l'Internet avec un accès Ethernet à 100Mb/s.
Pendant chacun des matchs, deux applications P2P IPTV différentes étaient lancées sur les noeuds (une application par noeud), ainsi que Windump qui était utilisé pour collecter le trafic. Le trafic récolté sur chaque noeud provenait uniquement de l'application mesurée et il n'y avait aucune autre source de trafic. A la fin de l'expérience, nous avions bien récolté quatre traces de trafic de quatre applications différentes : deux traces d'applications différentes pour chacun des deux matchs. Pour toutes les expériences de mesures, la bande passante consommée était relativement faible et n'a jamais dépassé les 10Mb/s. Les cartes Ethernet (100Mb/s) ont donc toujours été en mesure de collecter le trafic sans subir de pertes.
Au cours des ces expériences de mesures, tous les noeuds regardaient CCTV5, une chaîne de télévision chinoise disponible pour toutes les applications. Pendant les mesures d'un match, il était important de regarder la même chaîne de télévision afin que le comportement des utilisateurs et le trafic qu'ils génèrent soit globalement similaire dans chaque trace. Par exemple, durant les publicités, quelle que soit l'application, un utilisateur pourrait arrêter de regarder la chaîne ou même éteindre l'application avant de la relancer au début de la seconde mi-temps.

3.2. *Présentation des applications mesurées*

Pour notre campagne de mesures, nous avons choisi de mesurer le trafic de quatre applications P2P IPTV très populaires : PPLive, PPStream, SOPCast et TVAnts. Bien que ces applications soient disponibles gratuitement, elles sont propriétaires et leur code source n'est pas libre. Les détails d'implémentations de ces applications ainsi que les protocoles qu'elles utilisent sont donc inconnus. Cependant, toutes ces applications affirment utiliser des "protocoles de téléchargements simultanés" (*swarming protocols*) à la BitTorrent dans lesquels les flots vidéo sont divisés en pièces de don-

nées et chaque pair télécharge les pièces de données chez plusieurs pairs simultanément. Les pairs retrouvent les données grâce aux informations qu'ils s'échangent sur les données dont chacun dispose ou sur leur connaissance de pairs voisins. Grâce à ce trafic de signalisation, chaque pair découvre itérativement de nouveaux pairs, de nouvelles pièces de données disponibles et peut télécharger les flots vidéos de plusieurs pairs en même temps. Dans ces protocoles P2P, il y a donc deux types de trafic : le trafic de données vidéo où les pairs s'échangent entre eux les données et le trafic de signalisation où les pairs échangent les informations nécessaires pour obtenir les données.

Comme nous l'avons montré dans nos précédents travaux [SIL 07a], chacune de ces applications transporte les trafics vidéo et de signalisation différemment. PPStream utilise TCP pour tous les trafics (100%) alors que PPLive ajoute du trafic UDP pour certains trafics de signalisation (99.9% pour TCP). SOPCast utilise presque entièrement UDP (95%) tandis que TVAnts utilise les deux protocoles de transport TCP (75%) et UDP (25%). Par la suite, nous appellerons les applications qui utilisent principalement TCP les "applications TCP" (PPLive, PPStream et TVAnts) et "application UDP" celles qui utilisent majoritairement UDP (SOPCast).

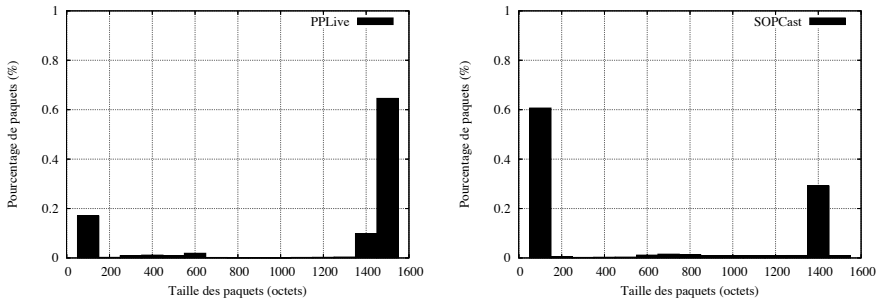
4. Méthodologie d'analyse

4.1. *Analyse de trafic multi échelle*

Nous analysons le trafic P2P IPTV à différentes échelles temporelles afin de caractériser ce trafic et d'en déduire les propriétés. Pour cela, nous calculons le spectre d'énergie du trafic à différentes échelles temporelles en utilisant une méthode basée sur les transformées en ondelettes [ABR 00]. La plus petite échelle temporelle analysée est l'intervalle de 20 millisecondes car nous avons observé dans nos traces que les temps d'inter arrivé des paquets sont très rarement inférieurs à cette valeur. Ainsi la granularité de nos observations est de de 20ms. Pour chacune des traces étudiées, nous avons compté le nombre d'arrivée de paquets contenant des données par intervalle de 20ms. Nous ne tenons pas compte des paquets ne transportant aucune donnée qui sont des paquets de gestion du protocole TCP (paquets de synchronisation ou acquittements).

Logscale Diagram Estimate [1] (LDestimate) utilise des transformées en ondelettes discrétisées et permet d'analyser le comportement d'échelle du trafic. LDestimate produit un diagramme logarithmique du spectre d'énergie du trafic. Pour tous les diagrammes logarithmiques (Fig. 3 et Fig. 4), l'axe des abscisses représente les octaves du trafic. Ce sont les échelles temporelles des arrivées des paquets. La partie la plus à droite des abscisses est ainsi relative aux grandes échelles temporelles et la partie la plus à gauche de l'abscisse correspond aux petites échelles temporelles. L'axe des ordonnés représente le spectre d'énergie du trafic. Un diagramme logarithmique doit être interprété de la façon suivante : une octave j est l'échelle temporelle du

1. http ://www.cubinlab.ee.unimelb.edu.au/∼darryl



(a) PPLive (b) SOPCast

Figure 2. *Distributions des paquets des applications étudiées.*

spectre d'énergie du trafic. La granularité du trafic analysé étant de 20 millisecondes, l'octave $j = 8$ signifie que l'échelle temporelle à laquelle on observe le trafic est $t = 2^8 * 20ms = 5.12s$.

LDestimate est un outil qui permet d'observer graphiquement les propriétés du trafic mesuré. Pour ces diagrammes : une bosse dans le spectre d'énergie indique un possible comportement périodique du trafic. Un spectre d'énergie constant signifie que le trafic pourrait être modélisé par des processus sans mémoire comme des processus de Poisson. Une croissance linéaire du spectre d'énergie témoignerait de phénomènes de dépendance à long terme du trafic (DLT).

En caractérisant le trafic des applications TCP, nous avons observé que leur spectre d'énergie et donc leur comportement multi échelle était similaire. En raison des limitations d'espace imposées par le format de cet article, nous présentons les résultats d'une seule des applications TCP (PPLive) et une application UDP (SOPCast). Les résultats pour les autres applications TCP (PPStream et TVAnts) sont similaires à ceux de PPLive et peuvent être consultés dans notre rapport technique [SIL 07b].

4.2. *Trafics de signalisation et vidéo*

Comme nous l'avons rappelé, les applications P2P étudiées génèrent deux types de trafic : du trafic de données vidéo et du trafic de signalisation. Les distributions des tailles des paquets des deux traces de trafic étudiées sont présentées sur la figure 2. Pour PPLive, 75% de ses paquets ont une taille supérieure à 1300 octets. La taille des autres paquets de PPLive est plus petite et inférieure à 100 octets. SOPCast compte seulement 30% de paquets dont la taille est supérieure à 1300 octets et la majeure partie de ses paquets (60%) a une faible taille ($\leq$ 100 octets). Ainsi, nous pouvons distinguer pour chaque trace deux ensembles de paquets : les paquets dont la taille est relativement faible ($\leq$ 200 octets) et les paquets dont la taille est importante (au moins supérieure à 1000 octets).

Le trafic vidéo, transportant des données volumineuses, devrait certainement être composé des paquets de tailles importantes ($\geq$ 1000 octets). De plus, le trafic vidéo sera sensible aux délais de traversée du réseau car les paquets vidéo devront respecter des instants de lecture strictes pour que les utilisateurs reçoivent une vidéo fluide

Tableau 1. *Statistiques de trafic pour les applications étudiées : trafic total et trafic vidéo obtenus en utilisant l'heuristique de filtrage.*

		Trafic Total		Trafic Vidéo	
Applications	Durée (secondes)	Vol. Total (Go)	#Pqt. Total ($x10^3$)	Vol. Video (Go)	#Pqt. Video ($x10^3$)
PPLive (TCP)	13 321	5.59	5 585	5.44 (97.3%)	4 060 (72.7%)
SOPCast (UDP)	12 198	4.77	10 425	4.12 (86.4%)	3 374 (32.4%)

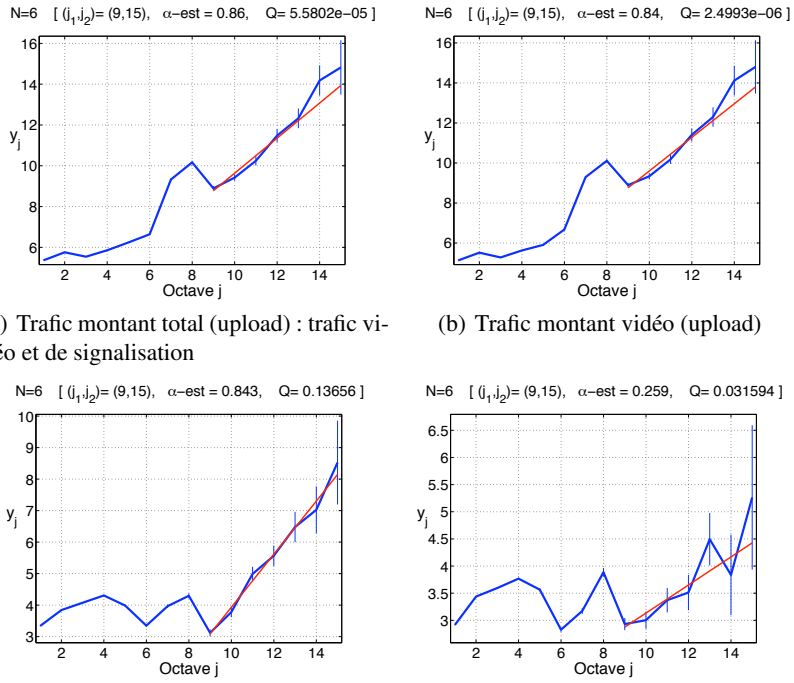
et de qualité. À l'inverse, les paquets du trafic de signalisation devraient être ceux de faible taille. Aucune contrainte de délais n'est à envisager pour les paquets de signalisation puisqu'ils sont simplement utilisés pour échanger des informations sur les pairs connus ou les données disponibles dans le réseau, mais pas pour transporter des commandes interactives comme pour des systèmes de vidéo à la demande comme Joost [JOO].

De par leur nature, les trafics vidéo et de signalisation n'ont pas les mêmes caractéristiques comme la taille de leurs paquets ou des contraintes temporelles. Il convient donc de séparer ces deux trafics dans notre étude afin de les analyser et évaluer leurs différents impacts sur le réseau.

4.3. *Heuristique de filtrage du trafic de signalisation*

En se basant sur les précédentes observations des distributions des tailles des paquets, nous avons choisi d'utiliser une simple heuristique [HEI 07] pour séparer le trafic vidéo du trafic total (vidéo et signalisation). Cette heuristique fonctionne de la façon suivante : Pour chaque flot, défini par les adresses et ports sources et destinations ainsi que le protocole de transport des paquets IP, on comptabilise le nombre de paquets dont la taille est supérieure ou égale à 1000 octets. Si un flot contient au moins 10 paquets dont la taille est supérieure ou égale à 1000 octets, alors ce flot est considéré comme un flot vidéo et l'on retire de ce flot tous les paquets de taille strictement inférieure à 1000 octets. Tous les flots qui ne sont pas considérés comme des flots vidéos sont retirés de la trace. A la fin, tous les flots restants constituent ce que nous définissons comme étant le "trafic vidéo".

Le tableau 1 présente les statistiques de trafic des traces étudiées comme leur quantité de données ainsi que leur nombre de paquets. Il présente également la quantité de données vidéo et le nombre de paquets vidéo obtenus après avoir filtré le trafic de signalisation des traces en utilisant l'heuristique de filtrage. Les rapports des données et paquets du trafic vidéo comparé au trafic total sont indiqués afin d'apprécier la justesse de l'heuristique. Par exemple, la trafic total de PPLive est composé de $5\ 585.10^3$ paquets comptant pour 5.59 Go de données et son trafic vidéo comprend $4\ 060.10^3$ paquets pour 5.44 Go. Le volume de trafic vidéo représente 97.3% du volume total et le nombre de paquets vidéo compte pour 72.7% du nombre de paquets du trafic total. L'heuristique de filtrage est bien parvenu à retirer les petits paquets et les sessions de signalisation sans pour autant affecter le volume de données vidéo et son nombre de



(a) Trafic montant total (upload) : trafic vidéo et de signalisation

(b) Trafic montant vidéo (upload)

(c) Trafic descendant total (download) : trafic vidéo et de signalisation

(d) Trafic descendant vidéo (download)

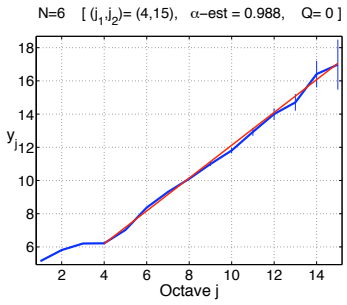
Figure 3. *Spectres d'énergie du trafic de PPLive. La granularité temporelle est de 20ms. (Ex : Octave $j = 8$ correspond à l'échelle temporelle $t = 2^8 * 20ms = 5.12s$).*

paquets. De la même façon, l'utilisation de l'heuristique pour SOPCast donne 86.4% de volume de trafic vidéo et 32.4% de paquets vidéos. La proportion de paquets vidéo paraît faible (32.4% du total) mais peut être expliqué en observant la figure 2(b). Pour SOPCast, plus de 60% des paquets sont des paquets de faible taille donc des paquets de signalisation à filtrer. De plus, nos précédents travaux [SIL 07a] ont montré que SOPCast avait généré une quantité importante de signalisation pendant ce match, réduisant de fait la proportion de paquets vidéo par rapport au nombre total de paquets.

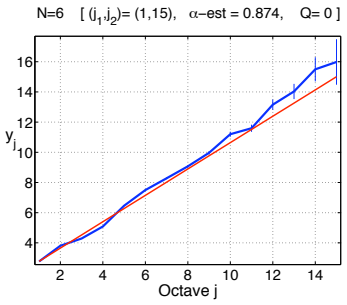
Dans cette section, nous avons montré comment séparer le trafic de signalisation du trafic total afin d'obtenir le trafic vidéo en utilisant une heuristique simple et efficace. Les résultats de l'analyse à différentes échelles temporelles de chacun de ces trafics sont présentés dans la section suivante.

5. Résultats

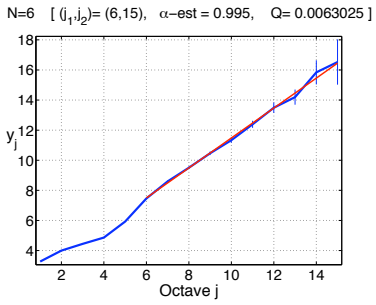
Pour chaque application, nous analysons le trafic en séparant les différentes directions du trafic : montant (*upload*) et descendant (*download*). Dans chacune des directions, nous séparons également le trafic vidéo du trafic total en utilisant l'heuristique présentée dans la section 4.3. Ainsi, chaque application est représentée par quatre digrammes logarithmiques : un pour le trafic montant total, un pour le trafic



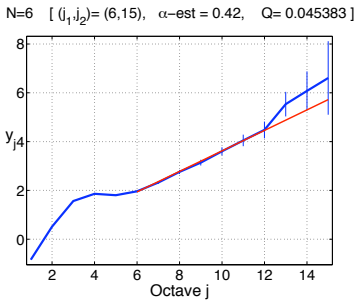
(a) Trafic montant total (upload) : trafic vidéo et de signalisation



(b) Trafic montant vidéo (upload)



(c) Trafic descendant total (download) : trafic vidéo et de signalisation



(d) Trafic descendant video (download)

Figure 4. *Spectres d'énergie du trafic de SOPCast. La granularité temporelle est de 20ms. (Ex : Octave $j = 8$ coorespond à l'échelle temporelle $t = 2^8 * 20ms = 5.12s$).*

montant vidéo, un pour le trafic descendant total et un pour le trafic descendant vidéo. La figure 3 présente les digrammes logarithmiques du spectre d'énergie du trafic de PPLive et la figure 4 ceux de SOPCast.

5.1. *Différences entre trafics TCP et UDP*

Pour PPLive, les deux spectres d'énergie des trafics montants sont similaires pour toutes les échelles temporelles (fig. 3(a) et fig. 3(b)). Les deux spectres d'énergie des trafics descendants se ressemblent jusqu'à l'octave $j = 9$ et diffèrent au delà (fig. 3(c) et fig. 3(d)). Les spectres d'énergie du trafic montant des applications TCP ((fig. 3(a) et fig. 3(b)) ne ressemblent pas à leurs spectres d'énergie du trafic descendant (fig. 3(c) et fig. 3(d)). Pour SOPCast, les quatre spectres d'énergie sont tous pratiquement identiques quelle que soit la direction du trafic ou sa nature (fig. 4). La légère différence pour le spectre d'énergie du trafic vidéo descendant (fig. 4(d)) sera expliquée dans la section 5.2.

Les spectres d'énergie de PPLive sont différents des spectres d'énergie de SOPCast. Cela signifie que les trafics TCP et UDP d'applications P2P IPTV n'ont pas les mêmes propriétés d'échelles et le même impact sur le réseau.

En étudiant les spectres d'énergie du trafic de PPLive (fig. 3), on peut observer une bosse dans tous ses diagrammes à l'octave $j = 8$ (5.12s). Cette bosse est cependant plus nettement définie dans le trafic montant (fig. 3(a) et fig. 3(b)) que dans le trafic descendant (fig. 3(c) et fig. 3(d)). Cette bosse dans les spectres d'énergie indique un possible comportement périodique du trafic à cette échelle temporelle quelle que soit sa direction ou nature. Les mécanismes de TCP pour transporter les données ou les retransmettre pourraient être à l'origine de ce comportement périodique. Or, ce comportement périodique est observé à l'échelle temporelle $j = 8$, soit 5.12s ; c'est une très longue période pour expliquer ce phénomène par la mise en oeuvre de mécanismes de TCP. De plus, nous ne tenons compte d'aucun segment TCP qui ne transportent pas de données (acquittements ou ouvertures de connexion). Les mécanismes de TCP ne sont probablement pas à l'origine de ce phénomène. Cette bosse pourrait provenir de la vidéo telle qu'elle est diffusée dans le réseau, mais SOPCast diffuse aussi de la vidéo et cette bosse n'a pas été observée dans ses spectres d'énergie. Ce phénomène périodique, propre aux applications TCP que nous caractérisons, ne semble pourtant pas provenir ni des mécanismes TCP ni de la vidéo diffusée. Actuellement, nous sommes toujours en train d'analyser ce phénomène pour en découvrir l'origine.

Concernant SOPcast (fig. 4), dans tous ses diagrammes on observe une croissance linéaire du spectre d'énergie quelle que soit la direction du trafic ou sa nature. Cette croissance linéaire indique une dépendance à long terme du trafic (DLT). Lorsque l'on détecte de la dépendance à long terme dans le trafic, cela indique que les conditions du réseau varient largement. Il devient alors difficile de respecter des paramètres de qualité de service (QdS) comme la garantie des délais de transmission ou de bande passante, paramètres nécessaires à la diffusion d'une vidéo de bonne qualité.

Toutes ces observations illustrent combien les protocoles de transport utilisés pour les applications P2P IPTV modifient les propriétés d'échelles du trafic généré et leur impact sur le réseau. Le trafic TCP présente un comportement périodique et le trafic UDP est dépendant à long terme. Ces résultats mettent en évidence le choix non trivial du protocole de transport pour ce type d'application puisqu'il est généralement admis que le trafic non élastique comme la vidéo devrait être transporté avec UDP. Pourtant, l'utilisation d'UDP entraîne la dépendance à long terme du trafic ce qui va dégrader la qualité de la vidéo reçue.

5.2. *Impact du trafic de signalisation*

Pour toutes les applications, quel que soit le protocole de transport qu'elles utilisent, leur spectre d'énergie de trafic montant vidéo ressemble à leur spectre d'énergie de trafic montant total. (fig. 3(b) et fig. 3(a) pour PPLive, et fig. 4(b) et fig. 4(a) pour SOPCast). L'utilisation de l'heuristique pour filtrer le trafic de signalisation n'a pas d'impact sur les spectres d'énergie des trafics montants et leurs propriétés. Pour les trafics descendants, les spectres d'énergie du trafic descendant vidéo sont différents de leur spectre d'énergie du trafic descendant total (fig. 3(d) et fig. 3(c) pour PPLive, et fig. 4(d) et fig. 4(c) pour SOPCast). Le filtrage du trafic de signalisation a modifié les spectres d'énergie du trafic descendant. Le trafic de signalisation a donc un impact

sur les propriétés d'échelles du trafic descendant mais n'a pas d'impact sur le trafic montant.

Cette observation est importante puisque le trafic de signalisation est nécessaire pour coordonner les transferts de données dans ces systèmes P2P. Pour des raisons de passage à l'échelle de l'Internet de ces systèmes, cette quantité de trafic de signalisation doit rester la plus faible possible afin de ne pas consommer trop de ressources réseau.

Dans chaque trace, le trafic de signalisation montant provient seulement du noeud contrôlé vers d'autres pairs sur l'Internet. Puisque nos noeuds ont de grandes capacités en bande passante car ils sont situés dans le réseau de notre campus, ils dupliquent des paquets vidéos à beaucoup d'autres pairs sur l'Internet. Comparé à l'important volume de données vidéo envoyé vers de nombreux pairs sur l'Internet, le trafic de signalisation montant envoyé par notre noeud ne représente qu'une faible partie du trafic montant total. À l'inverse, le trafic de signalisation descendant provient des nombreux pairs de l'Internet souhaitant bénéficier des importantes ressources offertes par les noeuds que nous contrôlons. L'autre partie du trafic descendant se limite au téléchargement de la vidéo. Le trafic de signalisation descendant provenant de nombreux pairs sur l'Internet compte ainsi pour une part importante du trafic total descendant. Cela explique pourquoi le trafic de signalisation a un impact sur le trafic descendant mais pas sur le trafic montant.

Plus généralement, les spectres d'énergie du trafic descendant des applications P2P IPTV étudiées (fig. 3(c), fig. 3(d) pour PPLive et fig. 4(c), fig. 4(d) pour SOPCast) sont complètement différents de leurs spectres d'énergie du trafic montant (fig. 3(a), fig. 3(b) pour PPLive et fig. 4(a), fig. 4(b) pour SOPCast). Ces observations sont d'autant plus pertinentes que nos mesures sont effectuées avec un accès symétrique à Internet. Les deux directions de trafic n'ont pas les mêmes propriétés d'échelles ni le même impact sur le réseau.

Les différentes propriétés de chacune des directions du trafic des applications P2P IPTV et l'impact significatif du trafic de signalisation sur le trafic descendant doivent être soigneusement pris en compte lors de la conception de modèles de trafic.

6. Conclusion

Dans cet article, nous avons étudié le trafic d'applications P2P IPTV à différentes échelles temporelles en utilisant une méthode de transformées en ondelettes du trafic. Cela nous a permis d'extraire des caractéristiques de ce type de trafic et d'en observer certaines propriétés et effets sur le réseau.

Notre analyse montre différentes propriétés des trafics TCP et UDP. Le trafic TCP présente un comportement périodique et l'utilisation d'UDP engendre des phénomènes de dépendance à long terme du trafic. Le choix d'UDP comme protocole de transport pour le trafic non élastique dans les réseaux P2P n'est alors plus si évident

puisque le phénomène de dépendance à long terme du trafic va dégrader la qualité de la vidéo reçue.
Pour toutes les applications, les propriétés d'échelles du trafic descendant sont différentes de celles du trafic montant. Le trafic de signalisation a également un impact significatif sur le trafic descendant mais pas sur le trafic montant. Cela met en évidence des problèmes de passage à l'échelle de ces systèmes P2P pour télécharger la vidéo. Les différences entre chacune des directions du trafic doivent être convenablement prises en compte lors de la conception de modèles de trafic.

En se servant de nos observations, nous concevons actuellement des modèles de trafic d'applications P2P IPTV. Des modèles de trafic réalistes sont en effet nécessaires pour obtenir des résultats fiables lors de simulations ou de modélisations de nouveaux systèmes P2P IPTV.

7. Bibliographie

[ABR 00] ABRY P., TAQQU M. S., FLANDRIN P., VEITCH D., « *Self-Similar Network Traffic and Performance Evaluation,* K. Park and W. Willinger, editors », chapitre Wavelets for the analysis, estimation, and synthesis of scaling data, Wiley, 2000.

[ALI 06] ALI S., MATHUR A., ZHANG H., « Measurement of Commercial Peer-To-Peer Live Video Streaming », *Proc. of Workshop in Recent Advances in Peer-to-Peer Streaming*, 2006.

[COH 03] COHEN B., « Incentives Build Robustness in BitTorrent », 2003.

[EDO] `http://www.edonkey2000.com`

[HEI 07] HEI X., LIANG C., LIANG J., LIU Y., ROSS K. W., « A Measurement Study of a Large-Scale P2P IPTV System », *IEEE Transactions on Multimedia*, 2007.

[JOO] `http://www.joost.com`

[MAG 07] MAGHAREI N., REJAIE R., « PRIME : Peer-to-Peer Receiver-DrIven MEsh-based Streaming », *Proc. of IEEE Infocom*, 2007.

[SIL 07a] SILVERSTON T., FOURMAUX O., « Measuring P2P IPTV Systems », *Proc. of ACM NOSSDAV*, 2007.

[SIL 07b] SILVERSTON T., FOURMAUX O., SALAMATIAN K., « Technical Report V2 Characterization of P2P IPTV Traffic : Scaling Analysis », 2007.
`http://arxiv.org/pdf/0704.3228`

[VEN 06] VENKATRAMAN V., YOSHIDA K., FRANCIS P., « Chunkyspread : Heterogeneous Unstructured End System Multicast », *Proc. of IEEE ICNP*, 2006.

[VU 07] VU L., GUPTA I., LIANG J., NAHRSTEDT K., « Measurement and modeling of a Large-scale Overlay for Multimedia Streaming », *Proc. of QSHINE*, 2007.

[YOU] `http://www.youtube.com`

[ZHA 05a] ZHANG X., LIU J., LI B., « On Large-Scale Peer-to-Peer Live Video Distribution : CoolStreaming and Its Preliminary Experimental Results », *Proc. MMSP*, 2005.

[ZHA 05b] ZHANG X., LIU J., LI B., YUM T. P., « Coolstreaming/Donet : A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming », *Proc. of IEEE Infocom*, 2005.

Monitorage et journalisation dynamiques des topologies dans les réseaux ad-hoc

Cristian Popi — Olivier Festor

MADYNES - INRIA Nancy Grand Est - Research Center[1]
615, rue du jardin botanique
54602 Villers-les-Nancy, France
{cristian.popi,festor}@loria.fr

RÉSUMÉ. *Nous étudions l'utilisation d'un service pair-à-pair basé sur des DHT (tables de hachage distribuées) pour le monitorage et la journalisation distribuée de l'historique de topologies sur un réseau ad-hoc. Notre approche tolère le fusionnement et le partitionement temporel d'overlay de monitorage. Elle permet ainsi à l'ensemble de fonctions de gestion (par exemple fautes, performance, configuration) de reconstruire un modèle de la topologie de réseau à tout instant de son historique. Nous présentons le protocole et le modèle de slot de temps utilisés par notre approche. L'architecture du système est décrite et son comportement est évalué par simulation.*

ABSTRACT. *We investigate the use of a Distributed Hash Table based Peer-to-Peer service to maintain the distributed monitoring history of an ad-hoc network topology. Our approach supports merging and splitting of monitoring overlays over time and enables various management functions (e.g. fault, performance, configuration) to rebuild a model of the network topology at any time in the past of its history. The protocol and time-slot models used are presented. The architecture of the supporting system is described and its performance is evaluated by simulation.*

MOTS-CLÉS : *réseau ad-hoc, monitorage, overlay*

KEYWORDS: *ad-hoc networks, monitoring, overlay*

1. Ce travail a été soutenu par le projet RNRT Airnet

1. Introduction

Un réseau ad-hoc est un réseau sans fil multi-sauts, dont les nœuds participants coopèrent pour acheminer des données ou découvrir des services. Un tel réseau est décentralisé et ne comporte pas nécessairement d'infrastructure préétablie. Les nœuds sont mobiles, et les liens entre eux instables.

La connaissance temporelle de la topologie d'un réseau ad-hoc est utile pour des nombreuses fonctions de gestion. La gestion des erreurs, par exemple, benefice de la disponibilité d'une telle information, puisque la détection des fautes dans le système est basée le plus souvent sur l'analyse des événements passés [QIU 06]. Dans les réseaux sans fil mobiles multi-radio, utiliser un historique de la topologie peut mener à diverses stratégies pour améliorer la QoS, en créant par exemple des algorithmes qui assignent des canaux aux interfaces radios d'un nœud [MAR 05]. Par conséquent, nous soulignons la nécessité et l'importance de maintenir un historique de la topologie du réseau dans un environnement dynamique et incertain tel que celui d'un réseau ad-hoc.

Nous proposons une approche distribuée d'observation de la topologie et supportant la journalisation de l'évolution de celle-ci sur une période de temps donnée au sein du réseau ad-hoc. Les nœuds participants rassemblent l'information de voisinage pendant de courts intervalles de temps, et la stockent aléatoirement à travers les autres nœuds dans le réseau. De plus, les nœuds ad-hoc se doivent d'offrir une interface pour la requête de topologie. Un gestionnaire peut alors interroger n'importe quel nœud du réseau pour obtenir les données de voisinage concernant un nombre défini d'intervalles de temps et ainsi reconstruire des images de la topologie physique.

Offrir un tel service sur un réseau ad-hoc est un véritable défi vue la mobilité des nœuds et les fluctuations des liens qui engendrent des changements fréquents de la topologie et donc un surcoût de messages et de traitement pour fournir en permanence une vue la plus complète et la plus juste possible de celle-ci à des applications de supervision. Afin que le plan de supervision soit adapté aux conditions spécifiques des réseau ad-hoc, celui-ci doit intégrer la nature dynamique de la topologie supervisée et doit impérativement être conscient de l'incertitude liée aux données qu'il va pouvoir collecter pour la représenter à un moment donné. De plus, le coût du plan de collecte d'information de gestion doit être maîtrisé ; un idéal inatteignable serait un coût nul. Notre approche est entièrement distribuée sur les nœuds participant au réseau ad-hoc. L'intergiciel de maintenance des données collectées s'appuie sur une architecture pair-à-pair structurée, maintenant les informations de topologies collectées dans le temps dans une table de hachage distribuée. Cette approche est intéressante pour (1) offrir un service d'accès à l'information de gestion à tout nœud participant qu'il soit gestionnaire ou agent, (2) pour distribuer la charge de stockage dans le temps sur le plus grand nombre de nœuds et de façon la plus équilibrée possible. Nous avons retenu Chord [STO 01] comme modèle pair-à-pair d'accueil, étendu avec un support de réplication des informations et un service de fusion/fission de tables de hachage. Nous y avons ajouté un algorithme de stockage des données par période temporelle. L'ar-

chitecture et les algorithmes spécifiques au traitement de données topologiques sont décrits dans cet article.

Le papier est structuré comme suit : la section 2 présente les travaux relatifs ainsi que des solutions sur lesquelles notre travail est construit. La section 3 présente l'architecture du système. La représentation des données topologiques et leur collecte sont décrites dans la section 4. La section 5 détaille le protocole de coopération entre les nœuds du système pour établir et maintenir l'information de topologie recueillie. Ensuite, l'algorithme pour construire la matrice de topologie à partir de l'information de topologie est détaillé (section 6). L'évaluation du framework proposé est réalisée dans la section 7. Celle-ci est suivie d'une conclusion.

2. Travaux Relatifs

DAMON [RAM 04] propose une architecture de surveillance distribuée pour les réseaux ad-hoc, basée sur l'existence de centres de dépôt, appelés puits, dans le réseau. Une organisation distribuée des puits est privilégiée par rapport à une organisation centralisée pour des raisons de passage à l'échelle. Les nœuds candidats pour jouer le rôle de puit sont des nœuds avec une mobilité réduite. Ceci fait des points d'accès, lorsqu'ils existent, des choix parfaits pour surveiller les nœuds clients mobiles qui leurs sont attachés, mais constituent des points de faille. Nous fournissons une approche entièrement distribuée ne présentant pas la vulnérabilité d'un unique point de collecte.

OLSR [CLA 03], peut en soi fournir la topologie du réseau. L'état partiel de lien est inondé par des nœuds participants, de sorte que les chemins les plus courts peuvent être calculés. Si les nœuds annoncent tout le voisinage, alors tous les nœuds dans le réseau peuvent avoir une pleine connaissance de la topologie du réseau. Le problème du maintient de cette information dans le temps reste présent. L'utilisation d'une approche de DHT (table de hachage distribuée) pour distribuer uniformément les données rassemblées avec un degré minimum de réplication (pour prévoir le cas de perte des nœuds) est une approche possible pour cette fonction. En outre, on peut déployer des réseaux ad-hoc avec une multitude de protocoles de routage développés ces dernières années, tels AODV [PER 03] ou DSR [JOH 07]. Si ils sont employés pour le routage, tirer profit des informations de topologie fournies par OLSR ne sera plus possible. Le fait qu'il y ait des protocoles de routage multiples renforce le besoin d'une solution qui soit indépendante de la couche de routage.

Nous proposons l'utilisation d'un overlay basé sur Chord [STO 01]. Dans la littérature de nombreux travaux ont porté sur l'adaptation de Chord aux réseaux ad-hoc [LAN 06, CAE 06, CAS 06, DAB 01a, DAB 01b]. Deux modèles ont notamment été proposés par [CHE 06] pour la composition dynamique de DHT : un modèle d'absorption et un modele par gateway. Une approche similaire, mais plus appropriée pour fusionner les tables de deux sous-partitions a été présentée dans [HEE 06]. Nous utilisons cette dernière dans notre approche.

3. Architecture de la solution

Notre solution de gestion comporte trois couches (fig. 1). La couche *application de gestion* utilise les services offerts par la DHT sous-jacente pour stocker/chercher l'information de topologie. Chaque nœud a la responsabilité de rassembler l'information de voisinage (comme présenté dans la section 4). Nous divisons le temps en intervalles discrets, que nous appelons des *slots*. Pour un slot de temps, que nous choisissons d'une longueur parametrable pour saisir suffisamment d'information de voisinage, et pour permettre également à un manager d'observer la dynamique du réseau, nous inférons le voisinage d'un nœud, et créons un enregistrement topologique, en associant l'information de voisinage au slot courant. Les enregistrements topologiques obtenus par chaque nœud, et pour chaque slot sont stockés dans la DHT recouvrant l'ensemble des nœuds contrôlés.

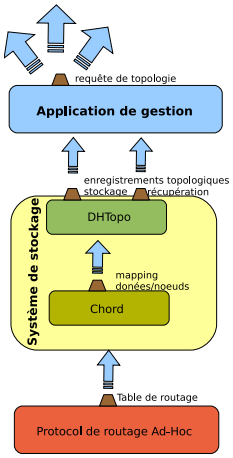


Figure 1. *Architecture du system de gestion avec les services offerts par chaque couche*

Le système de stockage est représenté dans la figure 1 par les couches *Chord* et *DHTopo*. *Chord* est responsable d'associer des identifiants aux nœuds, de construire et de maintenir les tables de routage overlay, ainsi que de la recherche des nœuds qui sont responsables d'un indentifiant donné. *DHTopo* emploie alors le service de recherche offert par *Chord* pour trouver le nœud sur lequel un enregistrement topologique est stocké. Le rôle de *DHTopo* est de stocker correctement les enregistrements topologiques concernant le même intervalle de temps pour différents nœuds. Une opération de *merge* est définie pour stocker les enregistrements des nœuds multiples sous la même clef Chord, pour le slot de temps courant, t_i. Le système de stockage fournit les services suivants à l'application de gestion : stockage sous la même clef des enregistrements de topologie (pour les mêmes intervalles de temps), et la récupération des enregistrements topologiques en une seule opération pour un slot de temps. La couche

application de gestion contrôle les slots de temps, organise l'information de voisinage et ajoute l'enregistrement ainsi obtenu dans la DHT en utilisant le service offert par la couche DHTopo. D'autre part, elle offre un service de requête aux gestionnaires souhaitant reconstruire un modèle de la topologie du réseau à un moment donné. Une requête peut être effectuée sur n'importe quel nœud de la communauté de gestion.

4. Collection et représentation des données

Puisque notre intérêt principal est l'information de la topologie, une représentation correcte devrait au minimum se composer d'une identification de chaque nœud, id, et d'un lien entre chaque paire de nœuds, $\mathcal{F}$, tels que, pour l'ensemble de tous les nœuds contrôlés $\mathcal{N}$, $\mathcal{F} : \mathcal{N}X\mathcal{N} \rightarrow \{'one_hop_neighbor', 'not_neighbor'\}$. $\mathcal{F}(x, y) =' one_hop_neighbor'$, si pour $x, y \in \mathcal{N}$, y est un voisin de x, c'est à dire, qu'il y a un lien unidirectionnel entre x et y, et $\mathcal{F}(x, y) =' not_neighbor'$ s'il n'y a aucun lien direct entre x et y.

Pour l'identification d'un nœud, nous avons retenu l'adresse MAC des interfaces sans fil de chaque nœud. Chaque nœud exécute un agent responsable d'intercepter des *beacons* venant des nœuds voisins. Nous organisons alors le voisinage de n'importe quel nœud n_i, comme une liste d'adresses MAC, $n_i \rightarrow \{n_{i1}, n_{i2}, ..., n_{il_i}\}, \forall n_i \in \mathcal{N}$, où l_i définit le nombre de voisins d'un nœud n_i.

Slots de temps

Nous divisons le temps en slots, t_i, de durée parametrable (dans nos simulation cet intervalle est fixé à 5 secondes). Chaque slot commence à un moment en temps (absolu) multiple de l'intervalle (e.g. 2008 January 13, 22 :24 :05). La topologie est capturée pour chaque slot. Pour tous les slots de temps t_i, les actions que l'agent effectue sur n'importe quel nœud n_k, sont les suivantes. Au début du slot de temps, il initialise sa liste de voisins ; le premier élément présent dans cette liste pour chaque slot est l'adresse MAC du nœud lui-même. Pour la durée entière du slot de temps, l'agent intercepte des beacons, les analyse, et si leur nombre est au-dessus d'un seuil défini, il ajoute l'adresse MAC du nœud envoyant les beacons dans la liste de voisinage ; si l'adresse est déjà dans la liste, toute autre beacon venant de cette adresse pour ce slot de temps est négligé. À la fin du slot l'agent crée l'enregistrement topologique, en datant la liste des voisins obtenue avec le slot de temps afférent, et l'ajoute à la DHT après attente d'une période de temps aléatoire (pour éviter de surcharger les nœuds responsables de l'hébergement des enregistrements topologiques pour ce slot de temps).

Chaque nœud, à la fin d'un slot de temps, appelle la primitive **put**(*timestamp*, *list*), fourni dans l'API de DHTopo. Puisque nous choisissons des slots de temps comme clefs en raison de leur unicité, les données de voisinage rassemblées depuis différents nœuds pour un certain slot de temps t_i seront mappées vers le même identifiant, et

stockées ainsi sur le même nœud de la DHT. DHTopo veillera à ce que les données venant de différents nœuds pour le même slot de temps ne soient pas écrasées (comme cela se produirait avec un fonctionnement standard de Chord), mais fusionnées dans une liste :

$$\{t_i, \{n_1, n_{11}, n_{12}, ..., n_{1l_1}\}, \{n_2, n_{21}, n_{22}, ..., n_{2l_2}\}, ...\} \quad [1]$$

et stockées sur le nœud dont l'identifiant est le plus petit identifiant supérieur a la clef obtenue à partir du hachage de t_i, et sur un certain nombre, r, de successeurs de la liste de successeurs du nœud de taille q, où $r \leq q$, comme représenté sur la figure 2.

Nous définissons un nombre maximum de slots de temps, T_N, qui peuvent être maintenus par le système. Nous contrôlons les T_N slots de temps de la façon First In First Out (FIFO), telles que les données stockées les plus anciennes seront toujours les premières à être remplacées par des données fraîchement rassemblées de voisinage. Pour une liste ordonnée de slots de temps t_0, t_1..., t_{T_N}, l'agent fonctionnant sur un nœud supervisé, a, fera ce qui suit :

– si le slot de temps courant, t_k, est inférieur à t_{T_N}, alors il pousse le tuple $\{t_k$, {a, neighborhood(a)}$\}$ dans la DHT et il maintient un $latest_slot$ égal à t_k ; ceci permettra de rejouer d'une manière ordonnée des données topologiques (par exemple $(dernier_slot, t_{T_N})$ suivi de $[t_0, dernier_slot]$) ;

– si le slot de temps courant, t_k, est t_{T_N}, alors il met le slot courant sur t_0 et il pousse $\{t_0$, {a, neighborhood(a)}$\}$ dans la DHT ; pour éviter d'ajouter l'information fraîche à la même liste contenant de vieilles information (pour le même slot de temps), au début de t_{i-1}, chaque nœud vérifie s'il est responsable de l'enregistrement topologique lié au slot de temps t_i ; s'il l'est, alors il vide la liste de l'équation 1 de son contenu.

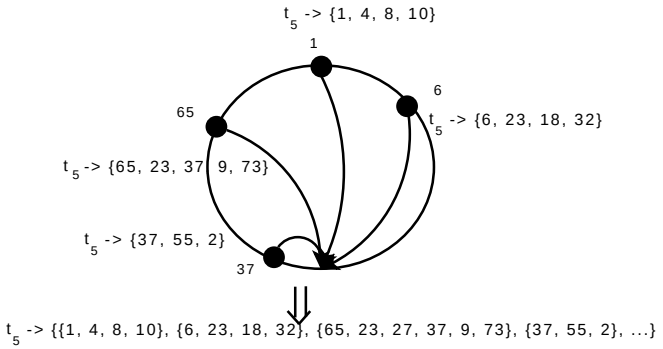


Figure 2. *Exemple de fusion des données de voisinage rassemblées dans une seule liste de type $\{n_k, n_{k1}, n_{k2}, ..., n_{kl_k}\}$, où l_k représente le nombre de voisins du nœud n_k ; le slot de temps t_5 se mappe au nœud ayant l'identifiant id=37 ; alors toute donnée de voisinage des nœuds 1, 6, 37, 65,... pour le slot t_5 sera stockée sur le nœud 37 dans une liste formée par la fusion des listes de voisinage individuelles*

5. Initialisation et maintenance de groupes supervisés

Chaque nœud de la communauté de gestion envoie périodiquement (chaque seconde) un message en broadcast avec un TTL=1, annonçant l'existence d'une organisation de gestion. Un TTL de 1 est suffisant pour faire participer tous les nœuds voisins (à une distance d'un hop) de la DHT, puisqu'après s'avoir rejoint le plan de monitorage, un nœud se comporte d'une façon similaire, prolongeant de ce fait la couverture de la DHT incrémentalement. Ce qui suit décrit ce qui se passe quand un nœud aléatoire, ne participant pas dans la DHT, reçoit l'annonce de l'existence de la DHT. Un nœud extérieur au groupe contrôlé qui reçoit l'annonce, accepte de se joindre et envoie un message de confirmation pour prouver qu'il est prêt à participer au plan de monitorage. Le nœud qui a fait l'invitation lance alors la procédure de rattachement du nouvel arrivé dans la DHT ; il hache l'IP du nœud invité, puis cherche un successeur pour le nouveau nœud ; il envoie alors de nouveau au nœud invité le successeur dans l'anneau de Chord. Parce-que les nœuds sont a priori synchronisé, le nœud peut commencer à collecter des informations de voisinage pour chaque slot et à les mettre dans la DHT.

Joindre deux DHTs séparées avec la même fonction (pour stocker l'information topologique) n'affectera pas les enregistrements topologiques dans les deux groupes fusionants, pour le même slot de temps, parce que l'opération de *merge* définie dans la couche *DHTopo* fusionnera les deux listes de paires $node-neighbors$. Ce mécanisme permet à un gestionnaire de faire une seule requête sur un des nœuds contrôlés après la fusion des deux partitions, pour obtenir une vue unitaire de la topologie du réseau pour un slot de temps correspondant à un moment où le réseau était divisé.

La procédure de bootstrapping est dans ce cas la suivante. Tous les nœuds écoutent pour une période de temps aléatoire (d'un maximum défini que nous avons fixé à 10 secondes dans notre simulation) pour de broadcasts annonçant l'existence d'une organisation de gestion. Après cette temps d'attente, chaque nœud qui a pas reçu un broadcast (et donc qui n'a pas rejoint un plan de monitorage) va créer une DHT, et va commencer à envoyer des broadcasts annonçant l'existence de sa DHT. Le nœud ne dépose des enregistrements dans la DHT qu'à partir d'un multiple de la durée du slot (5 seconds). Ce mécanisme assure qu'il aura un moins un nœud parmi tous les nœuds qui va bootstrapper le processus de collection/stockage de données topologiques. Intégrer deux DHTs revient donc au cas de la fusion décrit dans le paragraphe précédent.

En ce qui concerne la synchronisation, on fait l'hypothèse que les nœuds sont a priori synchronisés.

6. Algorithme pour la construction de la matrice de topologie

Un gestionnaire qui veut consulter la topologie pour un intervalle donné de temps, de t_i à t_j, doit faire $j-i+1$ opérations de *lookup* pour pouvoir rejouer la topologie du réseau pour cet intervalle. L'algorithme pour établir la matrice de topologie est décrit ci-dessous.

Input: $t_i..t_j$
Output: Topology matrices $\mathcal{M}_k$ for each time slot t_k

```
1  for t_k = t_i to t_j do
2      NeighborhoodList = lookup(t_k);
3      M_L={};
4      foreach node s ∈ the NeighborhoodList do
5          if s ∉ M_L then
6              add s to M_L;
7              foreach p ∈ M_L do
8                  M_k(s,p) = M_k(p,s) = 0;
               end
           end
9          foreach r ∈ NeighborList_s do
10             if r ∉ M_L then
11                 add r to M_L;
12                 foreach p ∈ M_L do
13                     M_k(r,p) = M_k(p,r) = 0;
                   end
               end
14             M_k(r,s) = 1
           end
       end
   end
```

Algorithm 1: L'algorithme pour construire et rejouer les matrices de topologie associées à une période $[t_i, t_j]$

$\mathcal{M}_k$ représente la matrice de topologie pour le slot de temps t_k et $\mathcal{M}_\mathcal{L}$ dénote l'ensemble des éléments index de la matrice carrée $\mathcal{M}_k$. À la deuxième ligne de l'algorithme, la liste $NeighborhoodList$ (ayant la forme de l'équation 1) est peuplée, en consultant l'entrée de la clef qui est responsable du slot t_k, et puis en obtenant les données de voisinage du nœud hébergeant la clef (la fonction *lookup* dans l'algorithme comporte en réalité deux appels de fonction : un pour rechercher les données topologiques, et l'autre pour les retirer). La ligne 3 initialise l'ensemble des indexes de la matrice, $\mathcal{M}_\mathcal{L}$. Puis, la matrice est construite, en parcourant toutes les sous-listes de $NeighborhoodList$, appelés $NeighborList_s$, qui contiennent les voisins de nœud s. Nous ajoutons s et ses voisins, r, dans $\mathcal{M}_\mathcal{L}$, et nous mettons $\mathcal{M}_k$ (r, s) à "1", pour montrer qu'il y a un lien direct entre r et s.

7. Évaluation du système

Nous avons implanté le framework de supervision distribué dans le simulateur OmNet++ [VAR 01] afin d'étudier sa faisabilité et son comportement. La couche physique

est simulée avec le modèle de propagation de disque-unité, avec une portée de transmission de 100m, et la couche liaison de données est basée sur le modèle IEEE 802.11 MAC. Au niveau du routage nous employons le protocole AODV. Ils font tous partie du framework existant INET, qui a été modifié pour contenir le protocole AODV. Au-dessus d'INET existe un framework de simulation pair-à-pair, OverSim, qui simule le protocole Chord, et la DHT construite sur Chord.

Dans la première série d'expériences nous avons fait bouger 10 et respectivement 100 nœuds dans un carré de 1000m x 1000m avec une vitesse aléatoire uniformément distribuée entre 15 et 20 m/s, et avec un temps d'attente aléatoire uniformément distribué entre 3 et 8 secondes. Pour la deuxième série d'expériences nous avons changé la vitesse des nœuds entre 5 m/s et 10 m/s. Pour chacune des ces deux séries d'expériences, nous avons choisi au hasard un slot de temps, t_x, et nous avons alors mesuré à chacun des 50 slots de temps suivants, t_y, l'exactitude du système, en comparant la matrice obtenue après une requête de topologie pour le slot de temps t_x, $M_{Q_{xy}}$, à la vraie matrice de topologie, M_{R_x}, sur la période t_x. L'exactitude mesure le pourcentage des entrées correctes dans la matrice de topologie obtenue par interrogation.

Nous avons ensuite comparé la précision de notre proposition, à la précision d'un système centralisé, dont un nœud, N, agit comme un collecteur de données topologiques. Pour chaque slot de temps, tous les nœuds retiennent la liste de leurs voisins (comme décrit dans la section 4). Le nœud collecteur est choisi aléatoirement parmi les nœuds de l'expérience. Ce nœud envoie des requêtes de topologie à chaque nœud, i (qui s'annoncent au préalable) pour chaque slot de temps. Pour chaque requête N attend une réponse contenant la liste des voisins pour le slot contenu dans la requête. Les requêtes sont envoyées périodiquement. Pour optimiser le nombres de messages envoyés dans le réseau, le collecteur transmet à chaque nœud, i, dans une seule requête, tous les slots pour lesquels i n'a pas répondu. La précision du système de polling est calculée comme décrit ci-dessus, c'est à dire, en comptant le nombre des entrées correctes de la matrice obtenue par polling, par rapport à la matrice de topologie réelle M_{R_x} pour un slot t_x.

Résultats

Les résultats pour deux vitesses différentes de mouvement de nœuds (5-10 m/s et 15-20 m/s) sont présentés dans la figure 3(a,b,c,d). Les figures montrent l'exactitude moyenne du système de supervision pour un slot de temps choisi au hasard, t_x, en changeant le niveau de réplication de stockage dans la DHT, pour un nombre de nœuds entre 10 et 100. L'exactitude augmente avec le nombre de slots qui passent après le slot de temps t_x et se stabilise après une certaine période de temps écoulée entre le moment de la mesure et celui de la consultation. La période dont le système a besoin pour se stabiliser est le temps qu'il lui faut pour agréger complètement les enregistrements de topologie pour t_x se retrouvant dans les DHTs sur la plateforme expérimentale ; l'incomplétude topologique de l'information est due au manque de contact initial entre des nœuds ou des zones des nœuds. C'est pourquoi, il faut autour de 150 secondes

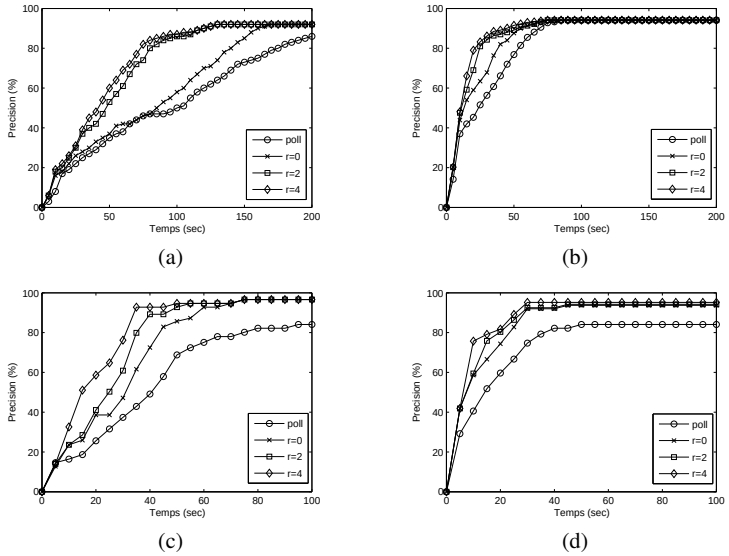


Figure 3. *Évaluation de l'exactitude moyenne du framework de supervision proposé vs. un système centralisé pour un nombre de 10 et 100 nœuds, qui bougent avec une vitesse aléatoire uniformément distribuée de 5-10 m/s et 15-20 m/s. Le niveau de réplication, r, de la DHT varie pour chacune des expériences. (a) 10 nœuds à 5-10 m/s ; (b) 100 nœuds à 5-10 m/s ; (c) 10 nœuds à 15-20 m/s ; (d) 100 nœuds à 15-20 m/s.*

(ou 30 slots de temps - fig. 3a) pour que l'exactitude du système soit élevée pour un nombre de 10 nœuds, mais elle prend seulement autour de 60 secondes (ou 12 slots de temps - fig. 3c) pour un nombre de 100 nœuds dans les mêmes conditions. Deux variables influencent l'exactitude du système : la vitesse avec laquelle les nœuds se déplacent et le niveau de la réplication des enregistrements topologiques dans la DHT. Une augmentation de chacun des ces deux paramètres diminue le nombre de slots de temps nécessaires pour collecter la topologie complète. Avec une vitesse accrue, les nœuds (ou les groupes isolés de nœuds) prennent moins de temps avant d'arriver à portée de transmission d'autres groupes de nœuds et fusionner leur DHTs ; cela améliore ainsi l'information topologique stockée.

En comparant les deux systèmes (dans la figure 3, *poll* représente la précision moyenne du système de polling), on conclut, qu'en moyenne, la précision du système DHT converge plus vite, vers une valeur élevée. Cela devient plus important avec l'augmentation du niveau de réplication dans la DHT. La raison est que dans le système de polling le collecteur doit "voir" tous les nœuds pour avoir l'ensemble de la topologie. Par contre, avec une DHT, il suffit d'avoir au moins deux nœuds qui détiennent des données topologiques complémentaires pour le slot t_x (e.g. de deux par-

titions différentes au moment t_x), pour pouvoir reconstruire toute la topologie. Plus la réplication augmente, plus il y a de chance que deux nœuds (ou groupes) se rencontrent et fusionnent leurs DHTs, contribuant ainsi à l'augmentation de la précision du système de monitorage.

Une exactitude parfaite est possible. Dans le simulateur cela signifie que la matrice de topologie réelle (celle donnée par le simulateur à la fin du slot de temps pour lequel nous interrogeons la topologie) est identique à celle obtenue par construction a partir de la liste de voisinage de chaque nœud dans la DHT. Pour être plus précis, les nœuds qui figurent dans la liste de voisinage du nœud N (ceux qui envoient un nombre de beacons supérieur au seuil défini) doivent toujours être voisins de N à la fin du slot de temps, quand le simulateur construit la matrice de topologie réelle.

8. Conclusion et perspectives

Nous avons proposé un framework de monitorage de topologie pour des réseaux ad-hoc, basé sur des tables de hachage distribuées. Le système stocke l'information topologique recueillie par les clients mobiles, à des intervalles de temps discrets, que nous avons appelés des slots de temps. Les instantanés de la topologie sont pris pour chaque slot de temps et sont stockés de manière distribuée, avec un équilibre équitable de charge sur chaque nœud dans le système. Le système peut alors être interrogé à tout moment pour connaître la topologie dans n'importe quel slot de temps de l'historique par un ou plusieurs gestionnaires auprès de n'importe quel nœud du réseau. Nous avons choisi de baser notre système de monitoring/journalisation sur un système pair-à-pair implantant une DHT avec réplication, car elle offre deux avantages importants : la résilience aux échecs de nœuds, et deuxièmement, le passage à l'échelle offert par Chord permet son application à un environmnet ad-hoc avec un nombre important de nœuds impliqués. Nous avons proposé une solution à la division provisoire du réseau. Celle-ci permet une construction de la topologie à partir de deux composants séparés, après leur fusion. Une fois que les deux sous-réseaux sont fusionnés, nous pouvons alors rejouer la topologie même pour la période de temps avant leur division.

En perspective, nous souhaitons étudier plus avant les influences des paramètres retenus pour notre étude (e.g. 5 s pour la durée d'un slot de temps), et faire de simulations à plus grande échelle sur plusieurs milliers de nœuds en analysant la surcharge due à notre approche. Ensuite, en s'appuyant sur notre système, nous projetons son utilisation pour collecter d'autres données telle la qualité des liens comme dans [KIM 06], et les quantifier sur une échelle (ex. de 1 à 5), pour pouvoir construire une matrice de topologie, qui au lieu de déclarer simplement qu'il y a un lien entre a et b, établit le niveau de fiabilité des liens.

9. Bibliographie

[CAE 06] CAESAR M., CASTRO M., NIGHTINGALE E. B., O'SHEA G., ROWSTRON A., « Virtual ring routing : network routing inspired by DHTs », *SIGCOMM '06*, ACM Press, 2006, p. 351–362.

[CAS 06] CASTRO M., O'SHEA G., ROWSTRON A., « Zero servers with zero broadcasts », *MobiShare '06*, ACM Press, 2006, p. 36–41.

[CHE 06] CHENG L., OCAMPO R., JEAN K., GALIS A., SIMON C., SZABO R., KERSCH P., GIAFFREDA R., « Towards Distributed Hash Tables (De)Composition in Ambient Networks. », *DSOM*, vol. 4269, Springer, 2006, p. 258-268.

[CLA 03] CLAUSEN T., JACQUET P., « Optimized Link State Routing Protocol (OLSR) », RFC Experimental n° 3626, Oct 2003, Internet Engineering Task Force.

[DAB 01a] DABEK F., BRUNSKILL E., KAASHOEK M. F., KARGER D., MORRIS R., STOICA I., BALAKRISHNAN H., « Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service », *HotOS-VIII*, Schloss Elmau, Germany, 2001, IEEE Computer Society.

[DAB 01b] DABEK F., KAASHOEK M. F., KARGER D., MORRIS R., STOICA I., « Wide-area cooperative storage with CFS », *SOSP '01*, ACM Press, 2001, p. 202–215.

[HEE 06] HEER T., GOTZ S., RIECHE S., WEHRLE K., « Adapting Distributed Hash Tables for Mobile Ad Hoc Networks », *PERCOMW '06*, IEEE Computer Society, 2006.

[JOH 07] JOHNSON D. B., MALTZ D. A., HU Y. C., « The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) », rapport, February 2007, IETF MANET Working Group.

[KIM 06] KIM K.-H., SHIN K. G., « On accurate measurement of link quality in multi-hop wireless mesh networks », *MobiCom '06*, New York, NY, USA, 2006, ACM Press, p. 38–49.

[LAN 06] LANDSIEDEL O., GÖTZ S., WEHRLE K., « A churn and mobility resistant approach for DHTs », *MobiShare '06*, New York, NY, USA, 2006, ACM Press, p. 42–47.

[MAR 05] MARINA M., DAS S., « A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks », *2nd International Conference on Broadband Networks*, vol. 1, Oct. 2005, p. 381–390.

[PER 03] PERKINS C. E., BELDING-ROYER E. M., DAS S. R., « Ad hoc On-Demand Distance Vector (AODV) Routing », RFC Experimental n° 3561, July 2003, Internet Engineering Task Force.

[QIU 06] QIU L., BAHL P., RAO A., ZHOU L., « Troubleshooting wireless mesh networks », *SIGCOMM Comput. Commun. Rev.*, vol. 36, n° 5, 2006, p. 17–28, ACM Press.

[RAM 04] RAMACHANDRAN K., BELDING-ROYER E., AIMEROTH K., « DAMON : a distributed architecture for monitoring multi-hop mobile networks », *IEEE SECON '04*, 2004, p. 601–609.

[STO 01] STOICA I., MORRIS R., KARGER D., KAASHOEK M. F., BALAKRISHNAN H., « Chord : A scalable peer-to-peer lookup service for internet applications », *SIGCOMM '01*, New York, NY, USA, 2001, ACM Press.

[VAR 01] VARGA A., « The OMNeT++ discrete event simulation system », *European Simulation Multiconference (ESM'2001)*, June 2001.

Gestion de la congestion

Vers un mécanisme de contrôle de congestion dirigé par les récepteurs pour le multicast

Vincent Lucas* — **Jean-Jacques Pansiot*** — **Dominique Grad*** — **Benoît Hilt****

** Laboratoire LSIIT (UMR 7005) - Université Louis Pasteur*
Boulevard Sébastien Brant - BP 10413
F-67412 Illkirch Cedex

{lucas,pansiot,grad}@dpt-info.u-strasbg.fr

*** Laboratoire MIPS (EA2032) - Université de Haute Alsace*
34 rue du Grillenbreit - BP 50568
F-68008 Colmar Cedex

benoit.hilt@uha.fr

RÉSUMÉ. *Pour généraliser le déploiement et l'utilisation du multicast, il est indispensable de disposer d'un mécanisme de contrôle de congestion. De plus, ce mécanisme doit être équitable avec TCP. Plusieurs méthodes dirigées par les récepteurs et utilisant un codage en couches ont été proposées. Ces méthodes sont extensibles aux très grands groupes, mais aucune n'est encore totalement satisfaisante. Dans cet article, nous proposons une nouvelle approche utilisant un codage en couches et s'inspirant de l'analyse de TCP. À partir de mesures expérimentales, nous avons dégagé certains paramètres influant sur l'efficacité et l'équité du contrôle de congestion. Nous proposons plusieurs mécanismes pour réduire significativement le temps de convergence vers un débit stable, et présentons des résultats encourageants en terme d'utilisation du réseau et d'équité avec TCP.*

ABSTRACT. *To generalize the deployment and the use of multicast, a congestion control mechanism is compulsory. Moreover, this mechanism must be fair with TCP. Several receiver-driven layered multicast congestion control scheme have been proposed. They are scalable to very large groups, but none of these are yet fully satisfying. In this article, we describe a new approach taking advantage of the layered scheme and of the analysis of the TCP congestion control scheme. Based on experimental results, we highlight parameters affecting the congestion control efficiency and fairness. Then, we propose several mechanisms to reduce significantly the time to converge to a stable rate, with interesting experimental results in term of throughput and of fairness with TCP.*

MOTS-CLÉS : *Contrôle de congestion, multicast, équité, TCP*

KEYWORDS: *Congestion control, multicast, fairness, TCP*

1. Introduction

Pour généraliser le déploiement et l'utilisation du multicast, un mécanisme de contrôle de congestion est indispensable, comme énoncé dans le RFC 2887 [HAN 00]. Nous nous intéressons aux mécanismes régis par les récepteurs, car utilisables à très grande échelle. La principale proposition est l'utilisation d'un codage en couches cumulatives. Ce principe demande à la source de subdiviser son flux en plusieurs couches ou canaux afin que chaque récepteur puisse indépendemment recevoir le débit qu'il souhaite en s'abonnant au nombre de canaux approprié. Plusieurs variantes ont déjà été proposées dont WEBRC [LUB 02][LUB 04] est sans doute la plus aboutie, notamment en ce qui concerne l'équité envers TCP.
Cette étude débute par l'état de l'art du contrôle de congestion pour le multicast à la section 2, avant d'analyser à la section 3 le comportement de TCP. La section 4 s'inspire de cette analyse pour proposer un premier protocole de contrôle de congestion, dont nous améliorons le temps de convergence dans la section 5. Enfin, la section 6 conclut sur les perspectives et les améliorations à venir de ce protocole.

2. État de l'art

Les méthodes de contrôle de congestion multicast qui exigent une signalisation entre récepteurs et émetteurs ne sont pas extensibles. De plus, les mécanismes utilisant un réseau actif sont actuellement difficilement déployables. Par conséquent, nous nous intéressons au contrôle de congestion multicast utilisant un codage en couches côté source et une gestion indépendante des abonnements aux canaux côté récepteur.

2.1. *Principes généraux côté source*

Les premières solutions, telles que RLM [MCC 96], RLC [VIC 98][RIZ 99] et FLID-SL [BYE 02], proposent de diviser le flux de la source en plusieurs couches ou canaux, chacun identifié par un groupe multicast. Le récepteur adapte son débit en s'abonnant au nombre de canaux adéquat. Cependant, se désabonner d'un groupe n'est pas immédiat et peut prolonger la congestion. Le récepteur n'est alors plus capable de déterminer si une congestion persistante est due au fait que le désabonnement n'est toujours pas effectif au niveau du lien congestionné ou s'il doit se désabonner d'un canal supplémentaire.
Pour corriger cela, FLID-DL [BYE 02] et WEBRC utilisent des canaux dynamiques. Le principe est que la source diminue continuellement le débit des canaux en fonction du temps. Le temps est alors divisé en cycles réguliers dont la durée est nommée *Time Slot Duration* (TSD). À chaque nouveau cycle, un canal est créé au débit maximum et le canal de débit le plus faible se termine et devient muet. Un récepteur peut ainsi réduire son débit en attendant simplement que les canaux diminuent d'eux-mêmes leur débit.

2.2. *Principe de WEBRC côté récepteur*

La méthode de WEBRC se rapproche de celle de TFRC [FLO 00b][HAN 03]. En effet, WEBRC estime la bande passante disponible suivant la formule [1] basée sur l'évaluation de Padhye [PAD 98] du débit moyen de TCP en fonction des pertes et du temps aller-retour nommé *Round Trip Time* (RTT) :

$$debit_{WEBRC} := 1/\left(artt * \sqrt{p} * \left(0.816 + 7.35 * p * \left(1 + 32 * p^2\right)\right)\right) \quad [1]$$

où p représente le pourcentage de pertes et $artt$ une estimation du RTT selon WEBRC. En effet, comme il n'existe pas de communication bidirectionnelle entre les récepteurs et la source, il est difficile d'estimer directement le RTT. Les récepteurs doivent donc se contenter d'une estimation qui pour WEBRC est équivalente au temps d'adhésion à un canal. Un calcul des abonnements à effectuer est réalisé à chaque nouvelle époque de 500 ms, qui est une subdivision du TSD. Finalement, cette décision est prise par comparaison entre l'estimation du débit avec la bande passante utilisée à la prochaine époque.
WEBRC est un protocole qui se veut équitable avec TCP. Cependant dans une étude précédente [LUC 06], nous avons montré l'inadéquation de l'utilisation du temps d'adhésion pour calculer l'estimation du RTT. De plus, même sur le réseau de tests de cet article, où le temps d'adhésion est représentatif du RTT, WEBRC présente encore des problèmes d'équité selon les caractéristiques du réseau, comme nous le verrons aux figures 4 et 6 lors des comparaisons avec les solutions que nous proposons.

3. Analyse de TCP

Dans tout l'internet, le principal contrôle de congestion utilisé est celui de TCP. C'est pourquoi, le protocole de contrôle de congestion pour le multicast que nous voulons élaborer doit être équitable avec TCP. Nous devons donc commencer par comprendre en détail comment fonctionne un TCP standard, par exemple New-Reno.
L'émetteur TCP utilise 2 variables, une fenêtre de congestion ou *Congestion Window* (CW) et un seuil de démarrage lent ou *Slow Start Threshold* (SST), qui sont respectivement initialisés à la taille d'un paquet ou *Segment Size* (SS) et à 64 Ko [JAC 88][STE 97]. L'algorithme de TCP comporte 2 phases et 2 événements :

– La première phase est le **démarrage lent**, conçue pour sonder rapidement la bande passante disponible. Chaque acquittement reçu augmente CW de la taille d'un paquet, ce qui correspond à un doublement par RTT (voir formule [2]), donc à une croissance exponentielle. Cette phase continue tant que $CW \leq SST$.

– La deuxième phase est l'**évitement de congestion**, conçue pour converger lentement vers le débit équitable. La source incrémente CW de la taille d'un paquet par RTT (voir formule [3]). Cette phase correspond à une augmentation linéaire et continue tant que $CW > SST$.

– Le premier événement est une **perte** détectée par exemple par la réception d'un acquittement en triple ou sélectif. Une perte est la manifestation d'une congestion et

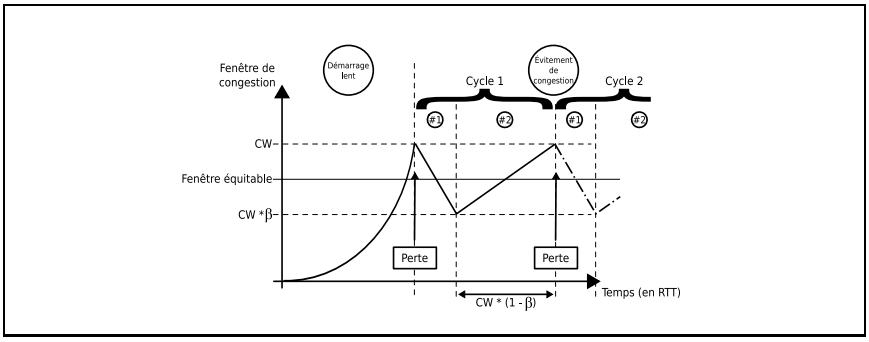


Figure 1. *Comportement de la fenêtre de congestion de TCP*

provoque la réduction de moitié de CW (voir formule [4]). De plus, SST est mis-à-jour et correspond à la nouvelle valeur de CW.

– Le second événement est l'**expiration du minuteur** enclenché au dernier acquittement reçu. Ce minuteur équivaut à un petit multiple du RTT moyen et permet de détecter les congestions sévères. À expiration, TCP reprend la phase de démarrage lent et SST est actualisé à la moitié de CW, qui est lui-même réinitialisé à la taille d'un paquet.

$$\text{Démarrage lent :} \quad CW := \frac{CW}{\beta} \qquad [2]$$

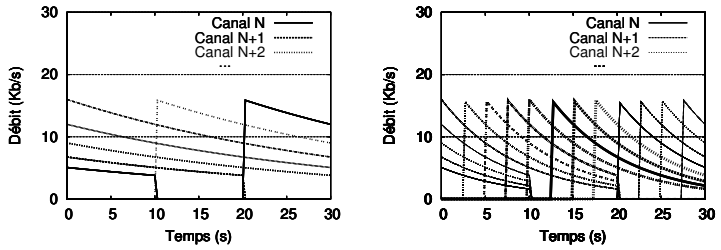
$$\text{Évitement de congestion :} \quad CW := CW + \alpha * \left(\frac{SS * SS}{CW}\right) \qquad [3]$$

$$\text{Perte :} \quad CW := CW * \beta \qquad [4]$$

$$\text{avec} \quad \beta := 0.5 \quad \text{et} \quad \alpha := 1$$

Quand CW atteint la taille correspondant à un partage équitable de la bande passante, le même débit est obtenu par chaque flux TCP de même RTT traversant le lien congestionné. Dans ce cas, chaque flux subit les cycles décrits dans [MAT 97] et représentés à la figure 1, composés d'une perte suivie d'une phase d'évitement de congestion. Chaque perte réduit CW de moitié (#1 de la figure 1). La phase d'évitement de congestion (#2 de la figure 1) nécessite $CW * (1 - \beta)$ RTT pour atteindre de nouveau la taille de CW et ainsi causer une nouvelle perte, signal du commencement d'un nouveau cycle. De plus, les analyses détaillées [FLO 00a][YAN 00] montrent qu'un protocole utilisant une augmentation additive et une diminution multiplicative avec une relation entre α et β telle que définie dans la formule [5] est équitable avec TCP.

$$\alpha = 3 * \frac{(1 - \beta)}{(1 + \beta)} \qquad [5]$$



(a) Les canaux de la source de WEBRC (b) Les canaux de la source avec réduction accélérée des débits

Figure 2. *Accélération de la réduction dynamique des canaux*

4. Contrôle de Congestion Multicast Inspiré de TCP (CCM-IT)

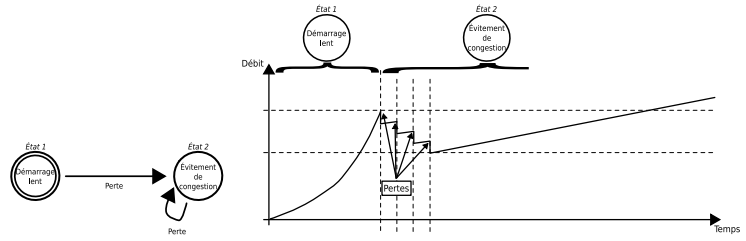
Notre première proposition est l'algorithme CCM-IT, basé sur l'analyse de TCP et sur la source de WEBRC. Cette partie présente une amélioration de la source, avant de se concentrer sur l'algorithme côté récepteur et d'évaluer cette proposition.

4.1. *Amélioration de la source*

Les canaux de la source de WEBRC diminuent trop lentement pour réagir convenablement à une congestion. En effet, cette diminution est de $1 - P = 25\%$ toutes les $TSD = 10$ secondes (voir figure 2(a)), avec $P = 0.75$ le facteur multiplicatif de diminution des canaux. Nous proposons d'utiliser une pente de diminution des canaux de la source plus importante tout en gardant une granularité de débit égale à celle de WEBRC. De plus, nous conservons TSD à 10 secondes pour éviter les problèmes liés au temps d'adhésion [LUC 06].
Par conséquent, nous proposons de passer le facteur multiplicatif de diminution P de WEBRC à P^K. Ainsi, la granularité inter-canal est toujours de P, mais ces derniers diminuent de P^K en un TSD (voir figure 2(b)). De ce fait les canaux réduisent leur débit K fois plus rapidement et au bout de TSD secondes il y a non pas 1 mais K canaux qui deviennent muets. Pour cette étude, nous définissons arbitrairement $K = 4$.

4.2. *Algorithme côté récepteur*

Les récepteurs ajustent leur débit en s'abonnant au nombre de canaux adéquat. Pour évaluer ce nombre, chacun compare le débit actuellement reçu à une estimation du débit équitable. Dès que le débit équitable dépasse celui reçu plus le débit de N canaux, alors le récepteur s'abonne à N canaux supplémentaires. Ainsi, il suffit de



(a) Diagramme états-transitions

(b) Principe de l'estimation du débit équitable

Figure 3. *Contrôle de congestion inspiré de l'analyse de TCP*

définir le calcul de l'estimation du débit équitable.
Le temps aller ou *One Way Delay* (OWD) est utilisé en remplacement du RTT. Pour cela, on suppose que chaque source et chaque récepteur a préalablement synchronisé son horloge par un moyen dont la précision est de l'ordre de la milliseconde, comme le protocole NTP ou la synchronisation par GPS. Cette synchronisation et un champ spécifique contenant l'heure ajouté dans l'entête permettent à chaque récepteur de calculer l'OWD_{actuel} pour chaque paquet reçu. De plus, nous utilisons l'OWD_{moyen}, qui est calculé selon la formule [6].

$$OWD_{moyen} \quad := \quad OWD_{moyen} * \beta + OWD_{actuel} * (1 - \beta) \qquad [6]$$

Pour estimer le débit équitable, nous allons nous inspirer de l'analyse de TCP et définir 2 phases et un événement (voir figure 3) :

– La phase de **démarrage lent** (*état 1* de la figure 3(a)) sert à découvrir la bande passante disponible. Comme pour TCP, nous utilisons une croissance exponentielle. À chaque paquet reçu, l'*Estimation du Débit Équitable* (EDE) augmente selon la formule [7]. Ainsi, l'augmentation est inversement proportionelle à la diminution du débit des canaux de la source. De plus, pour éviter les trop grandes rafales de pertes, cette augmentation est figée tant que l'estimation du $OWD_{actuel} > OWD_{moyen}$.

– La phase d'**évitement de congestion** (*état 2* de la figure 3(a)). C'est une croissance linéaire pour découvrir progressivement la bande passante disponible. EDE est actualisé à chaque réception de paquet en utilisant la formule 8.

– La réaction à une **perte** est une diminution multiplicative (voir formule [9]), proportionnelle à la réduction continue du débit des canaux de la source.

$$\text{Démarrage lent :} \quad EDE \quad := \quad \frac{EDE}{\beta} \qquad [7]$$

$$\text{Évitement de congestion :} \quad EDE \quad := \quad EDE + \alpha * \left(\frac{SS * SS}{CW} \right) \qquad [8]$$

$$\text{Perte :} \quad EDE := EDE * \beta \qquad [9]$$

$$\text{avec} \quad \beta := P^{\left(\frac{TIP}{TSD} * K\right)} \qquad [10]$$

$$\text{Ainsi que :} \quad \alpha := 3 * \frac{(1-\beta)}{(1+\beta)} \quad \text{et} \quad CW := \frac{EDE}{8} * OWD_{moyen}$$

Le calcul de α est le même que pour TCP, mais dans notre cas β est dynamique. En effet, β est déterminé par la réduction du débit des canaux de la source pendant le dernier *Temps Inter-Paquets* (TIP). Comme β est exprimé en fonction de TIP (voir formule [10]), cela signifie que β est actualisé pour chaque nouveau paquet reçu.

4.3. *Méthode de mesure*

Nous avons implémenté les différents algorithmes et les avons évalués en utilisant la maquette (voir figure 4(d)) qui utilise les protocoles de routage multicast MLD et PIM-SM. Cette dernière comporte un lien faible à 1 Mb/s entre des routeurs Linux qui nous permettent de configurer le temps de propagation et la taille des files d'attente. Pendant les différents tests, l'utilisation de la bande passante est toujours supérieure à 95%, ce qui nous permet d'étudier l'équité selon la formule [11].

$$EQ := \frac{debit_{multicast}}{debit_{TCP}} \qquad [11]$$

avec $debit_{multicast}$ le débit moyen par flux multicast

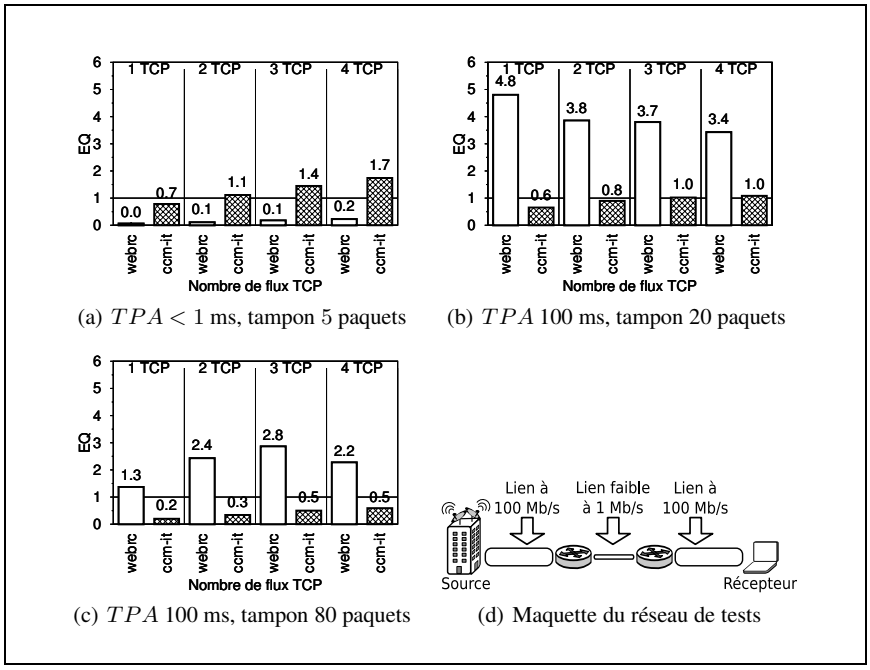
et $debit_{TCP}$ le débit moyen par flux TCP

Chaque test dure 10 minutes et révèle le comportement des protocoles en fonction du temps de propagation, de la taille des tampons et du nombre de flux TCP concurrents.

4.4. *Résultats*

Nous avons réalisé de nombreux tests en faisant varier la taille des files d'attente du lien faible de 5, 10, 20, 40, 60, 80 et 100 paquets, ainsi que le *Temps de Propagation Ajouté* (TPA) dans chaque sens de 0, 25, 50, 75 et 100 ms. Le OWD observé sera donc principalement la somme du TPA et du délai dans la file d'attente. Après avoir expérimenté toutes les combinaisons de ces paramètres, nous avons dégagé 3 comportements :

– Le comportement 1 apparaît quand les tampons des routeurs sont très petits, comme par exemple à la figure 4(a) avec un tampon de 5 paquets. Certes, CCM-IT semble équitable, mais le taux de perte d'environ 30% est beaucoup trop élevé. En ce qui concerne WEBRC, ce dernier présente un taux de perte d'environ 8% et il sous-utilise la bande passante. Par conséquent, les comportements de WEBRC et CCM-IT ont des dysfonctionnements qui montrent les limitations de ces méthodes quand les tampons sont sous-dimensionnés. Cependant, on peut considérer ce scénario comme



(a) $TPA < 1$ ms, tampon 5 paquets (b) TPA 100 ms, tampon 20 paquets

(c) TPA 100 ms, tampon 80 paquets (d) Maquette du réseau de tests

Figure 4. *Résultats de CCM-IT en fonction du nombre de flux TCP*

étant rare et fortement improbable au vu des configurations actuelles des routeurs qui par défaut utilisent bien plus d'espace mémoire pour les files d'attente.

– Le comportement 2 (voir figure 4(b)) apparaît avec des files d'attente de plus grande taille et OWD inférieur à environ 350 ms. Dans ces conditions, CCM-IT se montre équitable envers les flux TCP, avec un taux de perte inférieur à 4%. Pour WEBRC, le taux de perte est similaire, mais les flux utilisent 4 fois plus de bande passante que les flux TCP. Ce comportement montre que l'estimation de CCM-IT est plus précise que celle de WEBRC pour les configurations de réseaux les plus courantes.

– Le troisième comportement (voir figure 4(c)) apparaît quand OWD est supérieur à environ 350 ms. CCM-IT sous-utilise la bande passante alors que WEBRC est relativement équitable, même si cette équité varie en fonction du nombre de flux TCP concurrents. Dans les deux cas le taux de perte est toujours inférieur à 4%. En réalisant des tests de plusieurs heures dans les mêmes conditions, on peut observer que CCM-IT est finalement équitable avec TCP. Cela signifie que le problème vient du temps de convergence qui ici peut atteindre une heure pour un RTT d'une seconde.

CCM-IT est donc équitable pour les configurations de réseaux les plus courantes. Cependant, sa lenteur de convergence le rend difficilement utilisable en particulier pour le comportement 3.

5. Contrôle de Congestion Multicast avec Amélioration du Temps de Convergence (CCM-ATC)

La lenteur de convergence de l'algorithme CCM-IT est due à la phase d'évitement de congestion, créée pour atteindre en douceur le débit idéal. Ainsi, quand la phase de démarrage lent s'interrompt trop tôt, la phase d'évitement de congestion est utilisée alors que le débit est loin d'être équitable. Ce problème est similaire à celui de TCP pour les réseaux haut débit avec de longs délais. Pour améliorer CCM-IT, nous proposons CCM-ATC qui s'inspire de protocoles tels que BIC [LIS 04] et CUBIC[INJ 05].

5.1. *Analyse approfondie des cycles de pertes TCP*

Il faut donc déterminer la distance au débit équitable pour reprendre le cas échéant le démarrage lent et ainsi réduire le temps de convergence. Or, l'analyse de TCP montre que ce dernier est proche du débit équitable si le flux subit des pertes périodiques aux alentours du même débit. Ainsi, aux alentours du débit équitable :

– Une perte est attendue quand la période séparant deux pertes consécutives est écoulée. Cette période est estimée par $T := NB_RTT * OWD_{moyen}$.

– Une perte est attendue quand le débit utilisé est proche de celui auquel la dernière perte a été détectée. On peut définir que le débit R est proche du débit D de la perte précédente si $seuil_{min} < R < seuil_{max}$ avec $seuil_{min} := D{*}\beta^2$ et $seuil_{max} := \frac{D}{\beta^2}$.

5.2. *Amélioration du récepteur*

Avec ces nouvelles définitions nous pouvons ajouter 2 nouveaux états pour améliorer notre algorithme d'évitement de congestion comme décrit figure 5 :

– La phase de **démarrage lent** à l'*état 3*. Quand la durée depuis la dernière perte est trop longue, c'est-à-dire depuis plus de T secondes. Nous utilisons alors une croissance exponentielle pour converger rapidement vers le débit de la dernière perte.

– La phase d'**évitement de congestion** à l'*état 4*. Quand le débit est proche du débit reçu au moment de la dernière perte, nous ralentissons la croissance de notre estimation. Cela signifie que pour $seuil_{min} < EDE < seuil_{max}$, EDE augmente linéairement pour découvrir progressivement la bande passante disponible. Si $EDE > seuil_{max}$ et que le temps depuis la dernière perte est supérieur à $\gamma{*}T$ [1], alors le débit reçu lors de la dernière perte est trop loin du débit équitable, ce qui permet de reprendre le démarrage lent de l'*état 1* et ce jusqu'à la prochaine perte.

– Une **perte** provoque toujours un retour à l'*état 2* et réinitialise le minuteur T. De plus, si l'état précédent n'était pas l'*état 2*, nous recalculons les valeurs de $seuil_{min}$

1. Dans la suite, nous prenons $\gamma = 3$. Comme T est calculé en fonction de OWD_{moyen} qui peut être 2 fois plus petit que le RTT, il faut donc au moins que $\gamma \geq 2$ pour équilibrer cette approximation et $\gamma = 3$ permet d'être prudent avant de repartir en mode de démarrage lent.

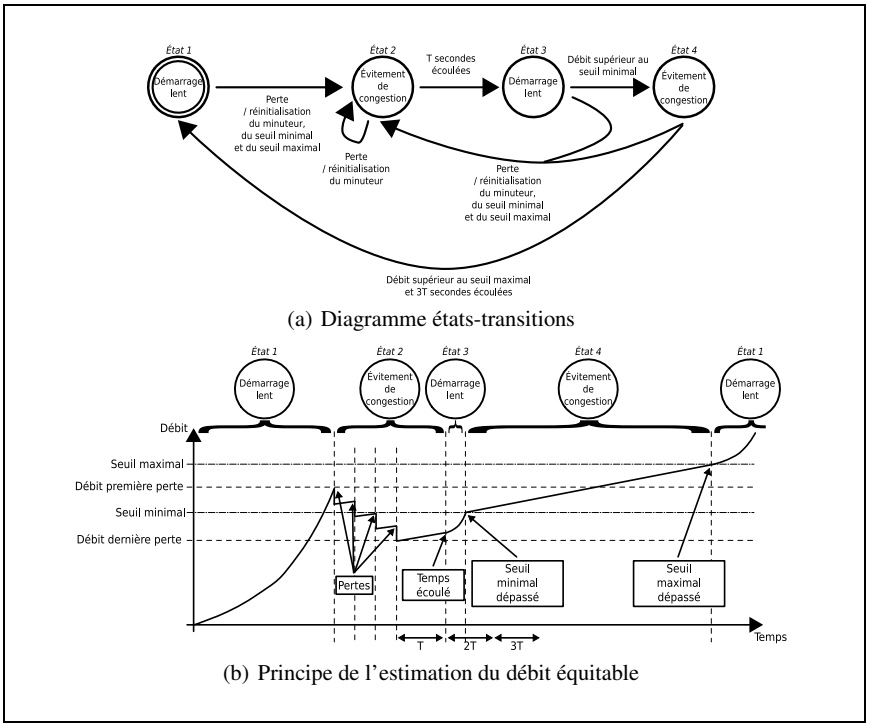


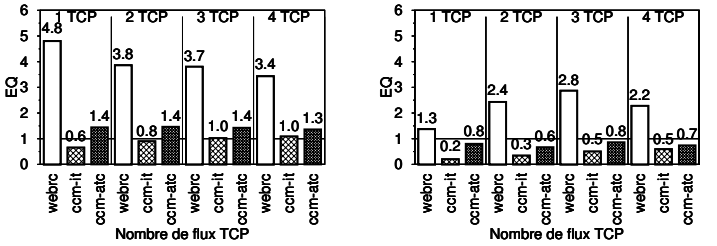
Figure 5. *Algorithme d'amélioration du temps de convergence*

et de $seuil_{max}$. De cette manière, en cas de rafale, ces seuils sont fixés à la valeur correspondant au débit lors de la première perte de la rafale.

D'une part, quand le débit est proche du débit équitable (*états 2 et 4*), cet algorithme croît linéairement, ce qui permet d'effectuer un partage équitable de la bande passante. D'autre part, quand le débit est éloigné du débit équitable (*états 1 et 3*), cet algorithme croît exponentiellement pour converger rapidement vers le débit équitable.

5.3. *Résultats*

Les mêmes tests que pour WEBRC et CCM-IT sont réalisés pour CCM-ATC afin de pouvoir comparer les 3 protocoles. Tout d'abord, le comportement 1 est identique pour CCM-IT et CCM-ATC en terme d'équité comme en terme de taux de pertes pour le cas des files d'attente sous-dimensionnées. Cependant, les tests de CCM-ATC (voir figure 6) ne présentent pas le comportement 3 de CCM-IT. Ainsi, CCM-ATC reste équitable pour tous les réseaux dont les files d'attente ne sont pas sous-dimensionnées tout en améliorant sensiblement le temps de convergence qui est divisé par un facteur



(a) TPA 100 ms, tampon 20 paquets (b) TPA 100 ms, tampon 80 paquets

Figure 6. *Résultats de CCM-ATC en fonction du nombre de flux TCP*

3 à 5 pour la figure 6(a) et converge en moins de 2 minutes là où CCM-IT en prenait plus de 60. Ces améliorations permettent donc de réduire le temps de convergence tout en stabilisant les flux au débit équitable. Ainsi, c'est l'auto-synchronisation de TCP qui permet de synchroniser indirectement les flux multicast.

6. Conclusion et perspectives

Cette étude nous a permis d'analyser le comportement de TCP et d'en dériver deux algorithmes de contrôle de congestion multicast. Le premier CCM-IT est fonctionnel tant que les tampons des routeurs ne sont pas sous-dimensionnés. Cependant, son temps de convergence très lent le rend inutilisable pour les réseaux où le RTT est supérieur à 350ms. Afin de résoudre ce problème, CCM-ATC propose de reprendre le mode de démarrage lent quand la probabilité d'être proche du débit équitable est trop faible. Cette probabilité est évaluée grâce à l'analyse des cycles de pertes que provoque TCP à un débit équitable. Finalement, les résultats de CCM-ATC montrent que ce dernier est équitable avec TCP tout en améliorant significativement le temps de convergence.

Ainsi ce protocole fonctionne dans les cas où TCP est majoritaire sur le réseau, ce qui est le cas le plus courant dans l'internet actuel. Cependant, si les flux TCP ne sont pas majoritaires, ces cycles de pertes ne sont plus respectés. Cela favorise le retour des flux multicast en mode de démarrage lent même si les récepteurs sont proches du débit équitable. La suite de nos travaux portera sur ce problème pour lequel il faudra pouvoir déterminer dynamiquement si les flux multicast sont majoritaires pour adapter l'agressivité du protocole. En outre, ces résultats doivent être complétés par de plus amples expérimentations impliquant des récepteurs ayant des OWD hétérogènes, ainsi que des tests avec des flux dynamiques. De plus, nous étudierons l'influence du facteur multiplicatif de diminution des canaux, ainsi que l'impact d'un tel contrôle de congestion sur la diffusion vidéo hiérarchique.

7. Bibliographie

[BYE 02] BYERS J. W., HORN G., LUBY M., MITZENMACHER M., SHAVER W., « FLID-DL : Congestion Control for Layered Multicast », *IEEE journal on selected areas in communications*, vol. 20, No.8, 2002, p. 1558–1570.

[FLO 00a] FLOYD S., HANDLEY M., PADHYE J., « A Comparison of Equation-Based and AIMD Congestion Control », 2000, www.aciri.org/tfrc/aimd.pdf.

[FLO 00b] FLOYD S., HANDLEY M., PADHYE J., WIDMER J., « Equation-based congestion control for unicast applications », *SIGCOMM 2000*, Stockholm, August 2000, p. 43-56.

[HAN 00] HANDLEY M., FLOYD S., WHETTEN B., KERMODE R., VICISANO L., LUBY M., « The Reliable Multicast Design Space for Bulk Data Transfer », RFC : Informational n° 2887, August 2000, IETF.

[HAN 03] HANDLEY M., FLOYD S., PADHYE J., WIDMER J., « TCP Friendly Rate Control (TFRC) : Protocol Specification », RFC : Standards Track n° 3448, January 2003, IETF.

[INJ 05] INJONG R., LISONG X., « CUBIC : A New TCP-Friendly High-Speed TCP Variant », rapport, feb 2005, PFLDnet 2005, Ecole Normale Supérieure - LYON - FRANCE.

[JAC 88] JACOBSON V., « Congestion Avoidance and Control », *ACM SIGCOMM '88*, Stanford, CA, August 1988, p. 314-329.

[LIS 04] LISONG X., HARFOUSH K., INJONG R., « Binary increase congestion control (BIC) for fast long-distance networks », *IEEE INFOCOM'04*, vol. 4, 7-11 March 2004, p. 2514-2524.

[LUB 02] LUBY M., GOYAL V., SKARIA S., HORN G., « Wave and Equation Based Rate Control Using Multicast Round Trip Time », *ACM SIGCOMM'02*, 2002.

[LUB 04] LUBY M., GOYAL V., « Wave and Equation Based Rate Control building block », RFC : Experimental n° 3738, April 2004, IETF.

[LUC 06] LUCAS V., PANSIOT J.-J., HOERDT M., « Influence du temps d'adhésion sur le contrôle de congestion multicast », FLEURY E., KAMOUN F., Eds., *CFIP 2006*, Tozeur (Tunisie), october 2006, Hermès, p. 115–126.

[MAT 97] MATHIS M., SEMKE J., MAHDAVI J., « The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm », *ACM - CCR*, vol. 27, n° 3, 1997.

[MCC 96] MCCANNE S., JACOBSON V., VETTERLI M., « Receiver-driven layered multicast », *SIGCOMM '96*, New York, NY, USA, 1996, ACM, p. 117–130.

[PAD 98] PADHYE J., FIROIU V., TOWSLEY D., KUROSE J., « Modeling TCP Throughput : A Simple Model and its Empirical Validation », *Proceedings of the ACM SIGCOMM '98*, 1998, p. 303–314.

[RIZ 99] RIZZO L., VICISANO L., CROWCROFT J., « The RLC multicast congestion control algorithm », 1999, http ://info.iet.unipi.it/ luigi/rlc99.ps.gz.

[STE 97] STEVENS W., « TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms », RFC : Standards Track n° 2001, January 1997, IETF.

[VIC 98] VICISANO L., RIZZO L., CROWCROFT J., « TCP-Like congestion control for layered multicast data transfert », *IEEE INFOCOM'98*, 1998.

[YAN 00] YANG Y. R., LAM S. S., « General AIMD congestion control », *ICNP '00*, Washington, DC, USA, 2000, IEEE Computer Society, p. 187–198.

Détection des événements de congestion de TCP

Pascal Anelli[1], **Fanilo Harivelo**[1], **Emmanuel Lochin**[2]

[1] *LIM - Université de la Réunion*
15 Avenue René Cassin, 97715 Saint Denis Messag 9 - France
{fanilo.harivelo, pascal.anelli}@univ-reunion.fr
[2]*Université de Toulouse - ISAE - DMIA*
10 av. Edouard Belin - BP 54032 - 31055 TOULOUSE Cedex 4
emmanuel.lochin@isae.fr

RÉSUMÉ. *Ces dernières années ont vu un intérêt croissant pour l'étude de la mesure des flots TCP. Plusieurs méthodes ont été proposées afin d'estimer de façon précise et rapide le taux de perte des paquets. Toutefois, et à notre meilleure connaissance, l'estimation et l'identification des événements de congestion TCP ne sont actuellement pas adressées. Suite à la standardisation de TFRC (*TCP Friendly Rate Control*), nous assistons à un intérêt croissant pour les protocoles de transport utilisant un contrôle de congestion de type* rate-based*. Aussi, ce type de contrôle de congestion repose sur la même base métrologique que celle de TCP. Dans ce contexte, la détermination précise des événements de congestion (CE) est un élément clé. En effet, ces derniers donnent une information essentielle pour le calcul d'un débit d'émission qui soit équivalent à celui de TCP dans les mêmes conditions. L'objet de cet article est de mieux identifier les CE de TCP afin de fournir une référence précise à ces nouveaux protocoles. Nous vérifions dans cette étude que TCP n'identifie pas de manière efficace les CE du réseau et proposons une méthode capable de mieux les déterminer. Cette détection est effectuée passivement à partir de la capture temps réel des paquets d'un flot TCP.*

ABSTRACT. *The problem of path loss rate estimation with TCP protocol has been actively studied by the research community. Several schemes have been proposed in order to accurately estimate the loss rate of a given path. These proposals are used for monitoring or metrology purposed. However, estimating and identifying the TCP congestion event is currently not addressed. While nowadays rate-based protocols received a particular attention from the research community, it seems important to better identify these congestion events in order to give an accurate reference to these protocols which aim at behaving in a fair manner with TCP. In this study, we propose a method able to identify a congestion event at the border of an autonomous system.*

MOTS-CLÉS : *TCP, Evénements de congestion, Algorithme de mesures*

KEYWORDS: *TCP, Congestion events, Measurments algorithm*

1. Introduction

Une des caractéristiques importantes du protocole TCP est sa capacité à adapter son débit d'émission en fonction de l'état de sa route dans le réseau. La source TCP fonctionne en boucle fermée sur un réseau qui est vu comme une boite noire avec l'objectif d'écouler le plus rapidement possible les données. Pour se faire, la source TCP ajuste son débit d'émission en fonction des retours du réseau. L'interprétation des retours est binaire : soit l'absence ou la présence de congestion ; la réception d'un paquet acquitté signifiant un retour positif. La perte d'un segment s'interprète comme un débordement d'une file d'attente et donc d'une congestion. Ce retour négatif indique à la source qu'elle doit diminuer son débit d'émission afin d'éviter l'effondrement du réseau (*congestion collapse*) et préserver son flot d'un taux de perte important. A ces considérations de performances s'ajoute une dynamique de débit qui tend vers un partage équitable des ressources entre les flots concurrents.

La période du contrôle, c'est à dire le délai entre l'action de la source et le retour, est imposée par le temps d'aller retour (RTT : *Round Trip Time*). En automatique, on qualifie cette caractéristique de système à retard. En effet, suite à l'émission d'un segment TCP, la source n'aura connaissance d'une éventuelle congestion dans le réseau qu'après un RTT plus tard. C'est donc sur une période équivalente au RTT que les congestions sont détectées par la source. Du point de vue de TCP, une congestion se produit à la première perte de paquet et dure l'équivalent d'un RTT (le temps nécessaire pour avoir un retour). Le nombre de pertes dans un RTT n'est pas pris en compte. Cependant le nombre total de pertes va avoir un impact sur la durée de la reprise qui dépend également des méthodes de reprise utilisées par TCP.

La notion d'événement de congestion (CE : *Congestion Event*) indique donc une fenêtre de données avec un ou plusieurs paquets perdus ou marqués ECN [FLO 03]. A noter que le terme "événement de perte" (LE : *Loss Event*) s'utilise pareillement et désigne la même notion ([FLO 07]). En d'autres termes, le retour négatif pour TCP n'est pas précisément les pertes de paquets mais l'événement de congestion qui peut aussi être notifié avec ECN. De façon générale, un CE constitue un retour négatif pour une source utilisant un contrôle de congestion en boucle fermée.

Une source TCP contrôle son débit d'émission au moyen d'une fenêtre de données. Lorsque la source détecte un CE, elle ramène la taille de sa fenêtre de contrôle de congestion à la quantité de données en cours de transit au moment de l'apparition de l'événement et la diminue de moitié. Puisque TCP est un protocole fiable, il effectue également la retransmission des données perdues. De manière simplifiée, on peut conclure qu'un CE se traduit par une diminution de la fenêtre de données en interne et par une retransmission visible de l'extérieure de la source TCP. Le CE pour la source TCP se termine lorsqu'elle reçoit l'acquittement de la donnée émise au moment de l'apparition de l'événement. Dans [FLO 04], cette donnée est appelée *"recover"*.

Cette notion de CE apparaît encore plus clairement dans la spécification de TFRC. TFRC est un contrôle de congestion s'appuyant sur une équation modélisant le comportement moyen de TCP dans le but d'être compatible avec celui-ci (principe dit

TCP-friendly présenté dans [HAN 03]). TFRC est notamment mis en œuvre dans la version CCID3 (*Congestion Control ID 3*) [FLO 06] de DCCP (*Datagram Congestion Control Protocol*) [KOH 06]. TFRC a formalisé la notion de CE utilisée naturellement dans TCP. L'équation de TFRC ne prend pas en compte la probabilité de perte en paramètre (très difficile à déterminer avec exactitude) mais un *loss event* équivalent au CE de TCP.

La détection de la congestion par TCP n'est pas exactement la congestion du réseau (une congestion du réseau correspond à un épisode de débordement de file d'attente de routeur [BAL 05]). Sur une échelle temporelle, TCP détecte la perte un RTT après son occurrence ou au délai d'expiration du temporisateur de retransmission (RTO : *Retransmission Time-Out*). Ceci dépend de l'importance du flot. En effet, un flot comportant peu de données (dit flot court) aura plus de difficulté à détecter une perte par *fast retransmit* car le nombre de segments à émettre après la perte peut être inférieur à 3 et donc empêcher la reprise. Dans ce cas, la détection de la congestion s'effectue un RTO après la perte. La congestion détectée par TCP fournit l'indication qu'il y a eu une congestion dans le réseau mais ne précise pas à l'instant de sa détection si elle est encore d'actualité et quelle a été la durée de la congestion du réseau. Comme la congestion peut persister à différentes échelles de temps [RAM 01], le début et la fin de la congestion du réseau ne peut être connu par un flot TCP. Cependant un CE du point de vue de TCP ne peut pas être plus long que la durée d'un RTT. La détection du CE représente une estimation de la période de congestion.

Les travaux présentés dans cet article visent à développer une méthode d'identification des CE par une mesure passive [1] en temps réel applicable à TCP. La détection des CE peut trouver une application dans la métrologie bien sur mais aussi dans le développement de fonction de notification de congestion externe à TCP ou pour des politiques de conditionnement de trafic TCP. Cet article est organisé de la façon suivante : la section 2 donne les hypothèses et détaille la conception de l'algorithme de détection des CE appelé *Implicit Congestion Notification (ICN)* ; la section 3 présente la campagne de validation ; enfin la section 4 donne la conclusion et les orientations futures de ces travaux.

2. Conception d'un détecteur d'événements de congestion

L'estimation du taux de pertes de paquets TCP est une information métrologique essentielle du transfert TCP. De précédentes contributions ont permis d'établir des méthodes de mesures passives du taux de pertes de bout en bout telles [BEN 02], [ALL 03], [MEL 02]. De manière générale, le principe de base de ces algorithmes consiste en l'écoute passive des numéros de séquence afin de d'identifier les ruptures de séquences d'un flot TCP. L'analyse effectuée des traces de ces études est alors différente. Dans notre cas, nous proposons d'étendre ces travaux de détection des pertes à la notion de CE. Pour cela on partira des algorithmes d'estimation de taux de pertes

1. sans action sur le trafic

utilisées dans ces algorithmes. Il est clair que le résultat obtenu ne sera plus un taux de perte mais plutôt un *Loss Event Ratio* défini comme étant le ratio du nombre de CE par rapport au nombre de paquets transmis. Dans la section suivante, nous présentons notre algorithme de détection de CE en donnant ses hypothèses de conception et la méthodologie d'interprétation utilisée.

2.1. *Hypothèses de conception*

L'hypothèse porte sur la localisation de la détection des CE effectuée depuis la source ou depuis un nœud immédiatement en aval de celle-ci (.i.e. avant le routeur de bordure). Dans ce cas, celui-ci doit être sur une route symétrique (les acquittements et les segments de données sont reçus par le nœud) et le RTT du flot mesuré dans le nœud est similaire à celui de la connexion TCP. Enfin, la contrainte de fonctionnement porte sur des mesures passives faites en direct sur le flot TCP et non sur l'analyse *a posteriori* des traces du flot TCP. Cette contrainte temporelle impose donc que la détection utilise uniquement les traces du côté émetteur.

Le principe développé par la méthode ICN consiste à déterminer les CE qui affectent un flot TCP par l'observation de son activité. L'observation des actions de TCP est faite à partir de la capture des segments du flot TCP. Comme TCP est un protocole de transport fiable, les données perdues sont retransmises. Un CE se déduit de l'observation des retransmissions dans le flot TCP. Cependant, toutes les retransmissions n'indiquent pas une perte et toutes les pertes ne signalent pas un CE. Quand le temporisateur de retransmission expire, TCP entame une procédure de reprise de type *go back* N, il se produit alors des retransmissions de segments déjà reçus. Il est aussi connu que TCP peut déduire à tort des pertes de paquets suite à un déséquencement opéré par le réseau (*network-reordering*) ou suite à une augmentation importante du RTT. Dans ce dernier cas, l'acquittement arrive trop tard pour désarmer le temporisateur de retransmission. Celui-ci ayant déjà expiré et retransmis inutilement les données. Ce genre de situation est qualifiée de mauvaise expiration du temporisateur (*spurious timeout*). Ces fausses indications de pertes sont traduites au niveau du contrôle de congestion de TCP par une réduction du débit d'émission et au niveau du contrôle d'erreur par des retransmissions inutiles. Les travaux [SAR 05, BLA 04] visent à rendre TCP résistant aux retransmissions inutiles. Enfin, toutes les pertes ne sont pas à prendre en compte de par la définition même d'un CE. Dans le cas de pertes multiples au sein d'une même fenêtre d'émission, seule la première perte sert à l'identification d'un CE. Les autres pertes de la fenêtre sont à négliger. Pour cela il faut pouvoir détecter la première perte et la taille de la fenêtre d'émission.

En résumé, la mesure des CE pour être la plus exacte possible doit :

– prendre en compte la taille de la fenêtre afin de distinguer les pertes déclenchant un CE des pertes faisant partie de l'événement lui-même ;

– identifier les retransmissions inutiles qui ne doivent pas être interprétées comme des pertes ;

– traiter les retransmissions multiples des mêmes données. Ce genre de situation se produit dans les congestions sévères. Dans ce cas, les CE se succèdent, on qualifiera par la suite cette situation de re-congestion.

2.2. *De l'interprétation d'un événement de congestion*

Comme présenté en introduction, un événement de congestion consiste en une ou plusieurs pertes se produisant sur une période de temps correspondante au RTT. Sachant qu'une fenêtre de données est émise dans un RTT, connaître la fenêtre revient à identifier les données émises pendant un RTT. Lorsqu'il y a une retransmission, le bas de la fenêtre pointe sur le segment retransmis. Le haut de la fenêtre correspond au numéro de séquence des données émises le plus important. Les retransmissions entre ces deux points sont interprétées comme faisant partie du même CE. Un CE commence lorsqu'une perte est identifiée et se termine quand le haut de la fenêtre de données au moment de la notification de congestion est acquitté.

La détection des retransmissions ne suffit pas pour identifier une perte. Il faut pouvoir s'assurer que la retransmission ne s'est pas produite suite à une erreur de TCP. Ce cas doit être pris en compte pour ne pas surestimer les CE dans les routes perturbées par les déséquencements ou les fortes variations de RTT. Le principe d'identification des retransmissions repose donc sur un délai d'attente T avant validation nommé par la suite délai de validation. Dans l'article [REW 06], les auteurs classent la retransmission comme inutile si elle est acquittée dans un délai inférieur à une fraction du RTT_{min} ayant pour valeur 0.75. La valeur de la fraction a un rôle important dans la détection des CE. Plus le délai de validation est proche du RTT et moins il y a de chance de valider un CE, l'acquittement pourra être reçu et la retransmission sera considérée comme inutile. L'algorithme ICN dans cette configuration peut sous-estimer le nombre de CE. A l'inverse, avoir un délai de validation très faible peut conduire à interpréter des retransmissions inutiles comme des pertes. En somme, la détection d'un CE s'effectue en deux temps. Tout d'abord, l'identification d'une retransmission puis la validation d'une retransmission pour une perte. Il est à noter qu'il existe une latence entre la congestion dans le réseau et sa détection au niveau d'ICN. En effet, la détection d'un CE apparaît à l'instant de la validation qui est en retard par rapport à l'instant de la retransmission et qui est elle-même se produit après un délai (de l'ordre de un à quatre RTT selon la méthode de détection de la perte). Le retard de notification du au délai de validation est la contre partie pour palier au manque de fiabilité de la détection des pertes par TCP.

Le RFC 2581 [ALL 99] complète la définition du CE en précisant que la perte d'une retransmission doit être prise comme une seconde indication de congestion. Les bornes de la fenêtre des données émises doivent s'ajuster au moment de la seconde indication de congestion. Ce cas demande cependant à prendre quelques précautions dans la validation du second CE. En effet, avec la version TCP New Reno [FLO 04], il existe plusieurs variantes dans la gestion du temporisateur de retransmission suite à un acquittement partiel. La variante recommandée dite *impatient* réarme uniquement

le temporisateur sur le premier acquittement partiel. L'objectif est de limiter la durée du *fast recovery* lorsqu'il y a des pertes multiples. En effet le *fast recovery* en l'absence de SACK (*Selective Acknowledgment*) [MAT 96] ne peut corriger qu'une perte par RTT. Lorsque le temporisateur expire, le segment est retransmis et la suite de la reprise s'effectue au moyen du *slow-start*. Du point de vue du réseau, cette retransmission ne signifie pas qu'il y a eu une perte dans le réseau. Il faut donc identifier cette retransmission afin de la classer comme une fausse indication de congestion. Lorsque ces événements se produisent, la route du flot TCP est en congestion ou sort de la congestion. Le délai de validation T doit pouvoir prendre en compte cette situation afin d'identifier avec exactitude les retransmissions inutiles.

2.3. *L'algorithme Implicit Congestion Notification (ICN)*

A partir de l'observation des segments de données et des acquittements, les CE de chaque connexion sont déduits au moyen d'une machine d'état qui identifie la phase du contrôle de congestion et qui classe les retransmissions comme utiles ou inutiles. Le contrôle de congestion de TCP réagit selon des notifications (*feedback*) binaires du réseau : absence ou présence de la congestion. La machine d'état utilisée par ICN (1) compte donc deux états induites par les notifications :

– l'état normal qui caractérise une connexion TCP où les transmissions sont sans pertes. Suivant l'algorithme dit de Karn [KAR 87], des mesures de RTT peuvent être faites en séquence dans cet état. A savoir que pour un segment émis, le délai est chronométré jusqu'à réception de l'acquittement correspondant. Une fois la mesure faite, le processus recommence pour le prochain segment transmis. La mesure du RTT sert au dimensionnement du délai de validation. Pour les connexions qui démarrent sur un réseau congestionné, les pertes de segments empêchent de prendre des mesures du RTT. La valeur d'initialisation est arbitraire et reprend celle du temporisateur de retransmission à savoir 3 secondes [PAX 00] ;

– l'état de congestion qui commence à la perte du premier segment d'une fenêtre de données. A chaque fois que l'algorithme ICN rentre dans cet état, un CE est comptabilisé pour la connexion TCP.

Deux états temporaires sont ajoutés entre les états des deux phases. Ces états visent à identifier les retransmissions inutiles à l'aide du délai de validation comme indiqué précédemment. Enfin, lorsque le haut de fenêtre (noté *recover* sur la figure 1) est acquitté, ICN repasse dans un état normal.

Dans le cas d'utilisation du drapeau ECN (*Explicit Congestion Notification*), les congestions ne sont plus déduites uniquement des pertes. Pour ce type de connexion, la méthode ICN doit être étendue pour analyser la signalisation de congestion dans les acquittements TCP. Autrement, ICN sous-estimerait le taux de CE affectant la connexion TCP. L'étude [MED 05] précise que ECN est utilisé par 2,1 % des hôtes en 2004. En conséquence, nous avons choisi de ne pas présenter ce cas que nous réservons pour une étude future. A noter qu'ICN ne traite en rien du contrôle d'erreur qui

reste du ressort de TCP. Si des améliorations dans la décision de retransmission sont introduites dans TCP, la méthode proposée étant indépendante du contrôle d'erreur, elle reste donc valide. En conclusion, ICN se veut général aux versions de TCP qui reposent sur un contrôle de congestion dont la notification négative est une perte.

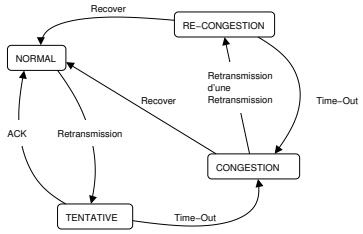


Figure 1. *Machine à état d'ICN*

3. Validation

Dans cette section, nous vérifions le principe de la méthode de détection des CE pour TCP et le délai de validation d'une perte. Notre objectif est d'estimer la fiabilité d'ICN. En d'autres termes, nous chercherons à déterminer quelle est l'erreur de notification d'un CE par rapport aux CE réels.

La validation du concept d'ICN est réalisée à l'aide de l'outil de simulation ns-2. La topologie utilisée est celle dite du papillon où n sources différentes sont connectées à n destinations différentes via un goulot d'étranglement entre deux routeurs. La bande passante et les temps de propagation de chaque lien sont respectivement de 1 Mbit/s et de 5ms. La bande passante du goulot d'étranglement peut être amenée à changer en fonction des études.Chaque source TCP src_i émet 400 paquets de taille égale à 1250 octets vers une destination dst_i. Le flot de paquets du couple (src_1, dst_1) forme la trace des segments TCP qui sont traités par ICN. Ce flot sera qualifié de flot de référence. Les $n-1$ autres couples source-destination génèrent un trafic transversal supplémentaire dans certains scénarios. Les n flots partagent les mêmes conditions de trafic (même profil de trafic et même RTT). Toutefois, ils démarrent en séquence dans un ordre aléatoire. La version de TCP retenue par défaut est celle dite New Reno [FLO 04]. L'objectif de ce modèle est de produire de la congestion pour le flot qui est analysé. L'intensité de la congestion est contrôlée par le nombre de flots en concurrence. Il est donc inutile d'agir sur le RTT des autres flots pour contrôler l'intensité de la congestion. La mesure de la précision de la détection des CE est faite à l'aide de l'erreur relative notée ϵ et qui se définie comme :

$$\epsilon = \frac{|N_{real} - N_{estimated}|}{N_{real}} \quad \text{avec } N \text{ le nombre de CE} \qquad [1]$$

La valeur $N_{estimated}$ s'applique au comptage fait par ICN ou par TCP. La mesure de référence N_{real} comptabilise les CE qui se sont réellement produits. Un CE réel se déduit selon les principes du CE d'ICN mais avec la connaissance certaine des pertes dans le réseau.

3.1. *Quel délai de validation ?*

Un élément important dans la prise de décision d'ICN réside dans son délai de validation T. Trois valeurs sont retenues pour son dimensionnement :

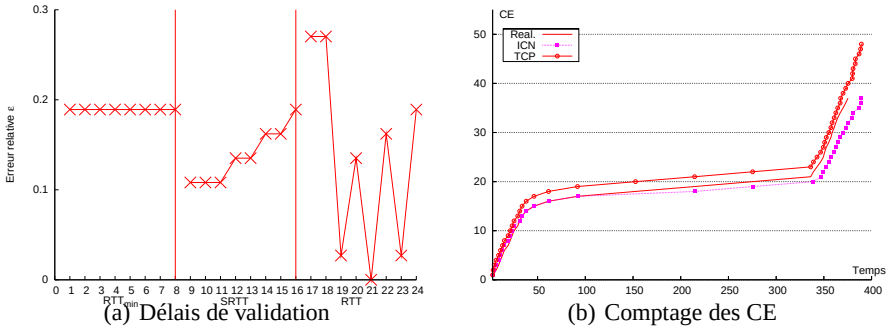
– RTT_{min}, celle-ci tend à attribuer une valeur faible ;

– $SRTT$ la moyenne exponentielle des mesures de RTT ;

– RTT issu de la dernière mesure, celle-ci tend à attribuer une valeur importante.

Ces valeurs sont ensuite pondérées par une fraction afin d'avoir une marge de sécurité. Cette marge vise à empêcher que l'expiration du temporisateur de validation du CE entre dans une période ou l'arrivée de l'ACK est probable. C'est à dire si l'acquittement de la retransmission déclencheur du CE est reçue pendant le délai de validation, le CE sera invalidé car la retransmission aura été classée comme abusive (à tort). La marge de sécurité traite donc de l'incertitude sur le RTT de la retransmission. Une marge de sécurité trop faible peut empêcher de détecter des CE et à l'inverse, avoir une marge de sécurité trop importante peut valider des CE sur des retransmissions inutiles. Les fractions testées sont $1, 0.75, 0.5, 0.25$. Dans le cas de re-congestion, le délai de validation sera testé avec ou sans pondération. La figure 2(a) présente l'erreur relative pour les 24 combinaisons possibles. Le scénario retenu porte sur 96 flots TCP. Ce qui assure une contention sévère des ressources. Les 8 premiers cas utilisent RTT_{min}, les cas 9 à 16 le $SRTT$ et les 8 derniers le RTT. Les valeurs de fractions sont évaluées dans l'ordre décroissant. Les cas numérotés par un nombre impair utilisent un délai de validation sans pondération pour les CE en situation de re-congestion. Dans ce scénario, le RTT_{min} est trop faible pour détecter les fausses notifications de CE. Avec $SRTT$, plus la pondération est faible, plus le nombre de faux CE augmente. Avec RTT, on remarque le rôle de l'absence de pondération pour valider les CE en situation de re-congestion. Ceci confirme notre analyse. Avec une pondération, le délai de validation est trop faible et des retransmissions inutiles déclenchent des faux CE. Pour le choix de la pondération, un minimum se dessine lorsque la fraction est de 0.5. Cette valeur donne un délai de détection assez rapide tout en préservant une marge de sécurité qui est équivalente au délai de validation. Par la suite, le délai de validation sera égal à $0.5 * RTT$ dans une situation normale et à RTT dans une situation de re-congestion.

La figure 2(b) montre l'évolution du comptage des CE pour le flot de référence dans le cas retenu (numéroté 21 sur la figure 2(a)). Au final, ICN a détecté 37 CE, TCP 48, alors qu'il y en avait 37 en réalité.



(a) Délais de validation (b) Comptage des CE

Figure 2.

3.2. *Analyse de la fiabilité*

Le première étude évalue l'erreur relative de ICN en fonction du taux de pertes de paquets. A cette fin, un modèle de simulation est constitué avec un seul flot TCP pour lequel des paquets sont perdus avec un taux variant de 10^{-3} à 10^{-1} selon un modèle de perte uniforme. Les autres caractéristiques du modèle de simulation reprennent celles données en début de cette section.

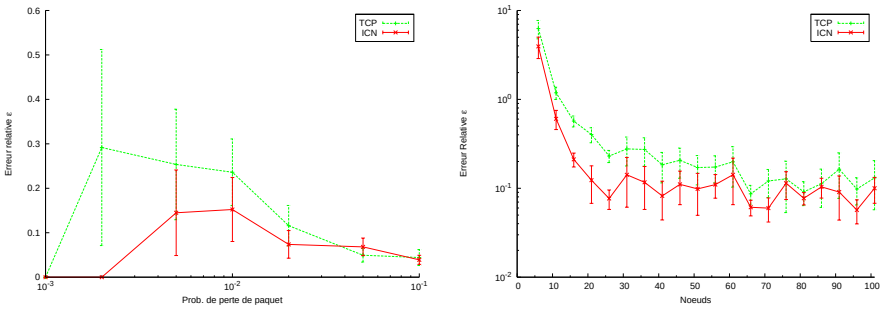
La figure 3(a) montre les résultats de cette étude pour la version de TCP New Reno. Afin d'apprécier la précision des résultats, 30 réplications différentes sont faites pour chaque taux de pertes. L'intervalle de confiance à 95 % de l'erreur relative est aussi représenté. A propos du calcul de l'erreur relative sur l'estimation des CE détectés par TCP, une précaution est à prendre quant à la détermination du nombre de CE de TCP. Ce comptage est déduit du nombre de réductions de fenêtre et du nombre de retransmissions quand la fenêtre de congestion a déjà la taille minimum. Dans ce dernier cas, la perte d'un segment retransmis ne peut diminuer la taille de la fenêtre puisque la fenêtre de congestion a déjà la taille minimum. Cette précaution est intégrée dans le comptage les situations de re-congestion. La figure 3(a) soulève plusieurs remarques :

– lorsque le taux de pertes est important, le nombre de CE est important, l'erreur relative a tendance à diminuer. A l'inverse sur des taux faibles, une erreur d'estimation entraîne une erreur relative importante du fait du faible nombre de CE. Le démarrage de TCP est souvent source de retransmissions inutiles car une congestion en phase de *slow start* est plus enclin à produire des erreurs multiples. La connexion TCP n'a pas encore acquis l'auto-synchronisation ;

– en moyenne, l'erreur relative d'ICN reste inférieure à celle de TCP.

La seconde étude vise à introduire une contention croissante avec le flot de référence. La contention se traduit par des périodes de congestion qui entraînent des pertes causées par débordement de la file d'attente. Dans ce modèle, le RTT varie en fonction du niveau de remplissage de la file et les pertes se produisent quand le RTT

atteint un sommet. La réalisation de ce modèle s'effectue en ajoutant un nombre de flots transversaux au flot de référence. La figure 3(b) montre l'évolution de l'erreur relative pour TCP New Reno et ICN lorsque le nombre de flots total varie de 6 à 101. En moyenne, l'erreur relative est importante lorsqu'il y a peu de flots à cause de démarrage de la connexion TCP comme expliqué précédemment. Plus la contention augmente, ICN converge vers TCP. La raison principale c'est qu'il y a de plus en plus de retransmissions inutiles sur expiration de temporisateur pour marquer la fin de la phase de *fast-recovery* de TCP New Reno. Ces retransmissions sont difficiles à détecter car l'ICN n'a plus de mesures de RTT dans l'état de congestion. Alors que le RTT a fortement augmenté. Le délai de validation n'est pas assez fort. Ce point reste encore à améliorer. Le problème ne provient pas d'un défaut de paramétrage d'ICN mais d'un défaut de mesure récente du RTT. Il est connu depuis [KAR 87] que les mesures de RTT en l'absence d'estampille temporelle ne peuvent être prise en compte lorsqu'un segment a été retransmis.



(a) Taux de pertes (b) Contention croissante

Figure 3.

3.3. *Retransmissions inutiles*

Bien que le modèle de simulation apporte quelques indications sur la méthode ICN. Il est difficile de reproduire la complexité et la diversité des situations qu'un flot TCP peut rencontrer. Le déséquencement introduit par le réseau est l'une de ces situations. Dans cette dernière, étude les pertes et les retards sont contrôlés afin de constituer un scénario avec un déséquencement du réseau produisant une retransmission inutile de TCP. La figure 4 montre l'émission et la réception d'un flot TCP New Reno dans lequel le segment 20 subit un retard suite un déséquencement du réseau et le segment 55 subit une double perte. Le segment 20 active un *fast retransmit* et chaque acquittement partiel déclenche une retransmission inutile. Dans cette situation particulière, ICN détecte la première retransmission qu'il classe comme inutile. Ce qui est intéressant dans ce cas, c'est que ICN est résistant à la fausse détection d'un CE sur les segments retransmis sur acquittement partiel [FLO 04]. Cependant ICN détecte les

2 CE qui se produisent plus tard. Au final, TCP aura souffert d'une fausse indication de CE en plus alors qu'ICN aura pu rester fidèle à la réalité. Cet exemple n'est certes pas exhaustif mais il montre le comportement d'ICN dans un cas connu.

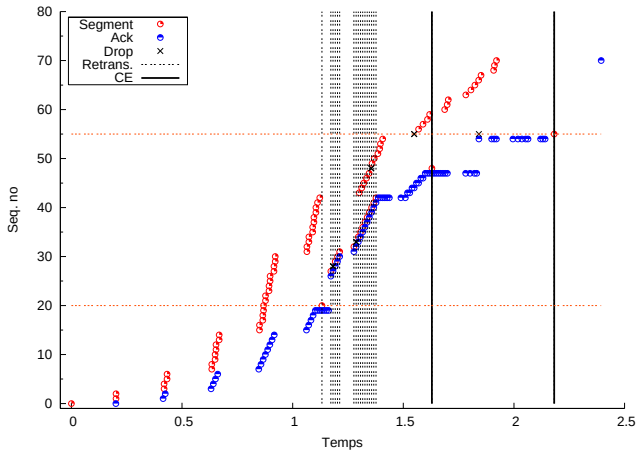


Figure 4. *CE identifié*

4. Conclusion

Dans cet article, nous avons présenté un algorithme qui détecte de façon plus fiable les événements de congestion de TCP. Cette détection est effectuée de manière passive en bordure du réseau. Cependant, cet algorithme doit encore être testé dans des conditions réelles pour apprécier ses caractéristiques de mise en oeuvre et ses capacités d'usage. Nous chercherons dans cette mise en oeuvre à quantifier le coût du délai inhérent à cette détection. Enfin, ICN gagnerait à être complété par un algorithme de discrimination de perte (*Loss Discrimination Algorithm*) afin de distinguer les pertes relatives à la congestion des pertes de corruption.

5. Bibliographie

[ALL 99] ALLMAN M., PAXSON V., STEVENS W., « TCP Congestion Control », Request For Comments n° 2581, 1999, IETF.

[ALL 03] ALLMAN M., EDDY W., OSTERMANN S., « Estimating Loss Rates With TCP », *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, n° 3, 2003, p. 12-24.

[BAL 05] BALI S., JIN Y., FROST V., DUNCAN T., « Characterizing user-perceived impairment events using end-to-end measurements », *International Journal of Communication Systems*, vol. 18, n° 10, 2005, p. 935–960, John Wiley and Sons Ltd.

[BEN 02] BENKO P., VERES A., « A Passive Method for Estimating End-to-End TCP Packet Loss », *Proc. of IEEE GLOBECOM*, novembre 2002.

[BLA 04] BLANTON E., ALLMAN M., « Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions », Request For Comments n° 3708, février 2004, IETF.

[FLO 03] FLOYD S., « HighSpeed TCP for Large Congestion Windows », Request For Comments n° 3649, décembre 2003, IETF.

[FLO 04] FLOYD S., HENDERSON T., « The NewReno Modification to TCP's Fast Recovery Algorithm », Request For Comments n° 3782, avril 2004, IETF.

[FLO 06] FLOYD S., KOHLER E., PADHYE J., « Profile for DCCP Congestion Control ID 3 : TRFC Congestion Control », Request For Comments n° 4342, mars 2006, IETF.

[FLO 07] FLOYD S., « Metrics for the Evaluation of Congestion Control Mechanisms », Internet Draft n° draft-irtf-tmrg-metrics-11.txt, octobre 2007, IETF.

[HAN 03] HANDLEY M., FLOYD S., PAHDYE J., WIDMER J., « TCP-Friendly Rate Control (TFRC) : Protocol Specification », Request For Comments n° 3448, janvier 2003, IETF.

[KAR 87] KARN P., PARTRIDGE C., « Improving round-trip time estimates in reliable transport protocols », *ACM Computer Communications Review*, vol. 17, n° 5, 1987, p. 2–7.

[KOH 06] KOHLER E., HANDLEY M., FLOYD S., « Datagram Congestion Control Protocol (DCCP) », Request For Comments n° 4340, mars 2006, IETF.

[MAT 96] MATHIS M., MAHDAVI J., FLOYD S., ROMANOW A., « TCP Selective Acknowledgment Options », Request For Comments n° 2018, octobre 1996, IETF.

[MED 05] MEDINA A., ALLMAN M., FLOYD S., « Measuring the Evolution of Transport Protocols in the Internet », *Computer Communication Review*, vol. 35, n° 2, 2005.

[MEL 02] MELLIA M., CARPANI A., CIGNO R., « Measuring IP and TCP behavior on Edge Nodes », *Proc. of IEEE GLOBECOM*, novembre 2002.

[PAX 00] PAXSON V., ALLMAN M., « Computing TCP's Retransmission Timer », Request For Comments n° 2988, novembre 2000, IETF.

[RAM 01] RAMAKRISHNAN K., FLOYD S., BLACK D., « The Addition of Explicit Congestion Notification (ECN) to IP », Request For Comments n° 3168, septembre 2001, IETF.

[REW 06] REWASKAR S., KAUR J., SMITH F., « A Passive State-Machine Approach for Accurate Analysis of TCP Out-of-Sequence Segments », *ACM Computer Communications Review*, vol. 36, n° 3, 2006, p. 51-64.

[SAR 05] SAROLAHTI P., KOJO M., « Forward RTO-Recovery (F-RTO) : An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol (SCTP) », Request For Comments n° 4138, août 2005, IETF.

Un modèle de décision pour l'auto-adaptation des protocoles de communication

Nicolas Van Wambeke — François Armando — Christophe Chassot — Ernesto Expósito

University of Toulouse - LAAS-CNRS
Laboratoire d'Analyse et d'Architecture des Systèmes
7, av. du colonel roche, 31077 Toulouse Cedex 4
{van.wambeke, armando, chassot, exposito}@laas.fr

RÉSUMÉ. *L'auto-adaptation des protocoles de communication est un enjeu majeur dans la conception des futurs services de l'Internet ambiant. Ceux-ci auront en effet à répondre aux besoins dynamiques d'applications complexes, mobiles et coopératives, distribuées dans des environnements réseaux hétérogènes, en partie sans fil et ad hoc, aux contraintes variables et évolutives. Dans ce nouveau contexte, dynamique à différents niveaux, les protocoles de bout en bout à architecture modulaire apparaissent comme un support privilégié pour l'auto-adaptation au contexte. Les contributions présentées dans cet article s'inscrivent dans cette approche et adressent le problème du choix d'architecture interne des protocoles auto configurables. Partant des propositions de protocoles à architecture modulaire issues des travaux de recherche actuels, nous proposons un modèle analytique pour guider le processus de décision, que nous confrontons ensuite, au travers d'un cas d'étude, à des mesures des performances réalisées sur une plateforme d'évaluation de protocoles à architecture configurable.*

ABSTRACT. *Self-adaptation of communication protocols is a major issue in the conception of future services for the ambient Internet. These protocols will have to provide adequate services to complex, mobile, cooperative application with dynamic requirements. These applications are distributed in highly heterogeneous environments, partially wireless and ad hoc with variable and evolutive constraints. In this highly dynamic context, end-to-end modular protocols appear to be well suited to support self-adaptation to the context. The contributions presented in this paper follow this approach and addresses the choice of internal architecture for modular protocols. We propose to guide adaptation by an analytical model. This model is then evaluated against simulation results.*

MOTS-CLÉS : *Auto-adaptation, protocoles auto-configurables, modéle analytique, Qualité de Service*

KEYWORDS: *Self-Adaptation, self-configurable modular protocols, analytical model, Quality of Service*

1. Introduction

Ces dernières années ont vu une modification radicale du paysage de la communication informatique : les applications distribuées dans l'Internet manipulent à présent tous les types de médias (e.g. audio, vidéo, ...), et impliquent des utilisateurs interagissant en temps réel dans le cadre d'activités potentiellement complexes (e.g. travail collaboratif) ; parallèlement, l'arrivée à maturité des technologies réseau sans fil préfigure une *mobilité généralisée des utilisateurs*. Dans un futur proche, l'information devrait être diffusable beaucoup plus largement par une utilisation massive de processeurs embarqués : l'Internet sera alors extrêmement *diffus, mobile et hétérogène*, au moins à sa périphérie, autorisant la mise en œuvre d'*activités coopératives complexes, mobiles et sans fil*, telles que les opérations d'intervention d'urgence dans des environnements peu ou pas pré équipés en infrastructure réseau.

Dans ce futur*Internet ambiant*, un enjeu de recherche majeur est de permettre aux applications et à leurs utilisateurs de disposer de *services de communication offrant des qualités de service (QdS) de bout en bout optimales* en fonction des ressources machine et réseau disponibles, dont les capacités et les performances ne sont pas connues à l'avance et sont potentiellement variables. Pour cela, il devient incontournable d'aborder la conception de protocoles de communication, et plus généralement de systèmes de communication, ayant des propriétés *d'auto-adaptation à la fois comportementales et architecturales*, vis à vis de la dynamicité des besoins applicatifs/utilisateurs, et de la variabilité des contraintes machine et réseau.

Dans ce cadre, les solutions pour l'adaptation proposées dans la littérature diffèrent en plusieurs points, relatifs aux objectifs ciblés (QdS, sécurité, ...), aux niveaux considérés (Application, Transport, ...), aux actions d'adaptation et à leurs propriétés. En particulier, les *protocoles à architecture modulaire* se présentent comme un support privilégié pour l'auto-adaptation [BRI 01, EXP 03a, MOC 05, WON 01]. C'est ce contexte que nous considérons dans cet article en ciblant plus spécifiquement les protocoles de bout en bout (niveau Transport).

Une difficulté majeure, sans solution générique à l'heure actuelle, est d'assurer la cohérence des choix d'adaptation à l'intérieur de chaque niveau et entre ces niveaux. Aborder le problème par le biais de modèles informels conduit à des solutions généralement peu évolutives (car associées à un contexte spécifique), et potentiellement sous optimales de part la complexité du problème. Face à ces limites, *guider l'adaptation par des modèles formels* est une approche prometteuse [FAR 06, LAN 04], en particulier étudiée au LAAS pour l'adaptation architecturale des protocoles de bout en bout, de niveau application, middleware et transport [CHA 06]. Les contributions présentées dans cette article prolongent ces travaux en abordant le problème du choix d'architecture interne des protocoles de transport dynamiquement configurables. Partant des propositions de protocoles à architecture modulaire issues des travaux de recherche actuels, nous proposons un modèle formel pour guider le processus de décision, que nous confrontons ensuite, au travers d'un cas d'étude, à des mesures réalisées sur une plate-forme d'évaluation des performances d'une composition de micro protocoles.

La suite de cet article est structurée comme suit : la section 2 propose un état de l'art des solutions d'adaptation, en particulier celles concernant les protocoles à architecture modulaire. La section 3 présente la contribution sur le modèle de décision pour l'*auto-adaptation*. La section 4 présente le cas d'étude et les mesures réalisées.

2. État de l'art

Les propositions de solutions pour l'adaptation au contexte sont multiples. Nous les avons étudiées et classifiées dans [CHA 06]. Nous résumons tout d'abord cette classification (section 2.1), puis nous détaillons les propositions d'architectures modulaires pour les protocoles de bout en bout auto configurables.

2.1. *Classification des propositions pour l'adaptation au contexte*

Les propositions pour l'adaptation au contexte diffèrent en plusieurs points, relatifs aux objectifs ciblés, aux niveaux considérés, aux actions d'adaptation, et aux propriétés de ces actions (voir Figure 1). Faute de place, nous ne détaillons ici que les propriétés de l'adaptation. Le lecteur est invité à consulter [CHA 06] pour disposer de l'état de l'art complet.

Comportementale vs. Architecturale. L'adaptation peut être guidée par des règles de transformation *comportementale* (ou algorithmique) ou *architecturale*.

L'adaptation est comportementale lorsque le comportement des composants est paramétrable ou modifiable. TCP et [AKA 04, WU 05] fournissent des exemples d'adaptation comportementale. La structure interne des composants n'est pas modifiée, ce qui facilite l'implantation mais limite la capacité d'adaptation vis à vis de l'ajout de nouveaux composants, non initialement prévus : pour cela, il est nécessaire de recompiler le composant et l'adaptation ne peut se faire en temps réel ; l'adaptation est dite *design time*.

L'adaptation est architecturale lorsque la structure interne du composant peut être modifiée. [EXP 03b, MIR 01, WON 01] ciblent par exemple la conception d'un protocole de bout en bout dont la structure interne est constituée de modules protocolaires composables dynamiquement en fonction du contexte. Nous détaillons plus avant cette approche dans la section (2.2). L'adaptation architecturale est dite *run time* lorsque la reconfiguration peut être effectuée durant la communication.

Vertical vs. Horizontal. L'adaptation peut concerner un composant local à une machine ou distribué entre plusieurs machines (typiquement, un module protocolaire).

Dans le premier cas, l'adaptation est qualifiée de *verticale*. Dans le second cas, l'adaptation est *horizontale*, et soulève des problèmes de synchronisation des entités adaptatives paires. [BRI 01] propose un premier modèle de protocole pour gérer cette synchronisation.

	Classification	
Objectifs	QdS, sécurité, optimisation de ressources, …	
Actions	Application	Contrôle de débit par encodage, contrôle d'erreur par FEC [WU 01] [YU 03]
	Middleware	Intergiciels réflexifs [KON 02]
	Transport	Contrôle d'erreur partiellement fiable [AME 94] Contrôle de congestion configurable [FLO 04] [AKA 04] Composition de micro protocoles [BRI 01] [WON 01] [EXP 03] [MOC 05] …
	Réseau	Routage orienté QdS [DAS 04] et [LEI 05] Configuration de services IP sécurisés [PER 04], …
	MAC	*Handover* vertical orienté QdS [BAL 04] [KAL 06], …
Propriétés	*Comportementale* [WU 01] [AKA 04] vs. *architecturale* [WON 01] [MIR 01] [EXP 03b] *Horizontale* [BRI 01] [MOC 05] [EXP 03b] vs. *verticale* [DAS 04] et [LEI 05] *Run time* vs. *design time*	

Figure 1. *Classification des solutions d'adaptation*

2.2. *Les protocoles à architecture modulaire*

Les protocoles à architecture dynamiquement composable ou modulaire sont basées sur la notion de *module protocolaire*. Il s'agit d'une *brique primitive* [HUT 91] résultant de la décomposition de la complexité d'un protocole en une succession d'actions élémentaires (i.e. contrôle de congestion, contrôle d'erreur, …). Dans ces architectures, un protocole résulte de la composition de plusieurs modules protocolaires concourant à la réalisation d'un service spécifique, adapté au contexte.

Ces architectures sont décomposables en deux approches : Coyote [BHA 98] et Cactus [BHA 99] suivent un modèle de gestion basé sur les événements, adapté aux traitements ponctuels tels que l'expiration de timer ; X-Kernel [HUT 91] et APPIA [MIR 01] sont régis par un modèle hiérarchique plus proche du modèle ISO et mieux adapté aux traitements systématiques (portant sur chaque paquet) [MEN 03].

Issue de nos travaux, l'architecture ETP [EXP 03a] suit une approche hybride tirant profit des deux approches précédentes par une décomposition des traitements en plusieurs plans (i.e. données, contrôle), optimisés en fonction des modules protocolaires qu'ils accueillent.

Ces nouvelles architectures sont de bons supports pour l'*auto-adaptation* mais dans l'état actuel, elles ne supportent qu'une adaptation architecturale *design-time* permettant de faire évoluer les fonctionnalités du protocole par ajout ou spécialisation de modules lors de la conception. Nos travaux actuels visent à concevoir des protocoles capables d'une adaptation architecturale *run-time*, c'est à dire pour lesquels la composition de modules est déployée dynamiquement en début de communication et peut être modifiée en cours de communication.

L'adaptation architecturale *run-time* soulève plusieurs problèmes :
- la synchronisation des entités paires est nécessaire pour une adaptation *horizontale* [BHA 99] ;
- le remplacement d'un module protocolaire par un autre (rendu possible par les évolutions récentes des langages objets) soulève des problèmes de performances [WON 01] ;

- enfin, le choix de la composition à instancier est un problème ouvert. L'objectif est de déterminer la composition de modules protocolaires qui réponde au mieux aux besoins de l'utilisateur et aux contraintes de l'environnement de communication.

C'est ce dernier problème que nous abordons dans cet article, en mettant l'accent sur l'adaptation des protocoles auto-configurables aux variations des contraintes du réseau.

3. Contribution

Face au problème précédent, nous présentons ci-après un modèle formel pour guider le processus du choix de composition. Un premier modèle, dit *de composition* a pour but de délimiter l'espace de recherche utilisé par un second modèle, dit *de décision,* qui a pour but d'automatiser le processus.

3.1. *Modèle de composition*

Le modèle de composition a pour but de définir les conditions de la validité d'un assemblage de modules protocolaires. Dans ce qui suit, ce modèle est instancié au regard des informations échangées entre les modules. En effet, bien que les modules soient indépendants (ils ont un fonctionnement propre), il est fréquent que leur fonctionnement requiert des informations produites par d'autres modules de la composition (sous la forme de variables partagées).

Définition : Soit A, un module défini par $A = [A^P A^R]$, avec :
A^P, le vecteur des publications de A tel que $A_i^P = \begin{cases} 1 & \text{si } A \text{ publie la variable } i \\ 0 & \text{sinon} \end{cases}$
A^R, le vecteur des besoins de A tel que $A_i^R = \begin{cases} 1 & \text{si } A \text{ requiert la variable } i \\ 0 & \text{sinon} \end{cases}$
Soit $C = \{(A)_1, \ldots, (A)_k, \ldots, (A)_N\}$ une composition de N modules partageant de l'information via M variables partagées. Il vient alors :

$$C \text{ est dite } \textit{valide} \Leftrightarrow \begin{cases} \sum_{k=1}^{N} (A_i^P)_k \leq 1 & \forall i \in [1..M] \\ \sum_{k=1}^{N} (A_i^P)_k > 0 & \forall i \text{ tq } \sum_{k=1}^{N} (A_i^R)_k > 0 \end{cases} \quad [1]$$

La première condition permet de garantir l'absence de conflit dans la publication des variables, dans le sens où elle empêche deux modules d'une même composition de publier la même variable, et en garantit ainsi la cohérence. La seconde condition permet de vérifier que toute variable déclarée requise par au moins un module est publiée par un autre. Elle permet de garantir que tous les modules instanciés dans C ont à leur disposition les informations dont ils ont besoin pour fonctionner.

Ce modèle peut être étendu pour prendre en compte d'autres éléments (i.e. interdépendance des modules, ordre d'instanciation, …) par ajout de nouvelles contraintes au système [1].

3.2. *Modèle de décision*

Le modèle de décision vise à modéliser le processus de choix d'une composition parmi P dans le but de maximiser l'efficacité perçue par l'utilisateur (déclinable par exemple en termes de QdS, niveau de sécurité, ...).

Définition : Soit A^E le vecteur *d'efficacité* associé à un module protocolaire A. On note A_c^E l'évaluation obtenue pour le module protocolaire A dans le contexte réseau c. Si deux modules protocolaires (A et B) sont évalués dans le même contexte et que A se montre plus efficace que B (au regard de critères de succès préalablement établis), alors l'inégalité $A_c^E > B_c^E$ est vérifiée.

Ainsi, pour un contexte donné c, le problème du choix d'une combinaison de modules protocolaires donnant les meilleures performances consiste à choisir $C = \{(A)_1, ..., (A)_k, ..., (A)_N\}$ comme étant la solution au problème d'optimisation suivant :

$$max \quad \left(E = \sum_{k=1}^{N} (A_c^E)_k\right) \qquad [2]$$

$$\text{sous} \begin{cases} \sum_{k=1}^{N} (A_i^P)_k \leq 1 & \forall i \in [1..M] \\ \sum_{k=1}^{N} (A_i^P)_k > 0 & \forall i \text{ tq } \sum_{k=1}^{N} (A_i^R)_k > 0 \\ min(\sum_{i=1}^{M} (\sum_{k=1}^{N} (A_i^P)_k - \sum_{k=1}^{N} (A_i^R)_k)_i) & \end{cases}$$

La solution C retenue sera celle qui, tout en maximisant l'efficacité, respectera les contraintes de validité du modèle de composition.

Dans le système d'équation ci-dessus, une nouvelle contrainte apparaît, qui permet, dans le cas où deux solutions C ont la même efficacité, de choisir celle qui a le coût le moins élevé en termes de variables publiées sans être requises par d'autres modules (i.e. minimisation de l'utilisation mémoire pour l'embarqué). Cette troisième équation illustre comment le modèle peut être enrichi par l'ajout de contraintes permettant de le spécialiser en fonction de critères propres au domaine d'application, ou émanentes de l'utilisateur (i.e. le module de chiffrement *3DES* doit être activé).

Note : Dans cette première phase de nos travaux, nous faisons l'hypothèse que les modules sont indépendants en performance, c'est à dire que la présence ou l'absence d'un module B n'influe pas sur les performances d'un module A. Cette hypothèse sera reconsidérée dans nos travaux futurs.

4. Étude de cas

Afin d'illustrer l'intérêt du modèle, nous présentons ci-après une étude comparative de différentes compositions valides dans l'ensemble de modules $D = \{M_1, M_2, M_3\}$. L'objectif est d'optimiser la QdS dans un contexte de réseau sans fil IEEE 802.11e.

4.1. *Présentation fonctionnelle des modules*

Module M_1 : Le module *QoSOptimizer* a pour but d'optimiser la QdS perçue par l'utilisateur pour un flux multimédia en fonction du niveau de congestion du réseau et des besoins en QdS des flux transportés. Ce mécanisme opère en supprimant de façon sélective certains ADUs[1] lorsque le débit d'émission maximum autorisé (reflet du niveau de congestion) devient trop faible pour que le flux complet soit transmis sans engendrer une augmentation inacceptable du délai.

Module M_2 : RateControl est un module spécifique réalisant la mise en forme du trafic selon un mécanisme de contrôle de congestion suivant l'algorithme TFRC *(TCP Friendly Rate Control)* [HAN 03]. TFRC est un contrôle de congestion, qui à partir d'une équation, permet de calculer le débit maximum autorisé sur la base du taux de pertes et du débit observés par le récepteur d'un flux. Suivant la spécification de TFRC, tout le trafic dépassant la limite calculée est retardé en conséquence.

Module M_3 : Le module *QdS Adapter* est spécifique aux environnements IEEE 802.11e. Ce standard définit quatre catégories d'accès au canal de communication qui ont chacune une probabilité différente de pouvoir émettre des données afin de privilégier les flux les plus prioritaires vis à vis du trafic de fond moins prioritaire. L'utilisation du *QdS Adapter* permet de classifier les paquets en leur assignant une catégorie d'accès en fonction de la nature applicative des flux véhiculés suivant les recommandations du standard IEEE 802.11e. La vidéo se voit ainsi attribuer un débit supérieur au trafic moins prioritaire.

4.2. *Présentation analytique des modules*

La communication entre modules se fait par le biais de trois variables partagées : *frameType, maxSendingRate et availableAccessCategories* respectivement publiées par les modules M_1, M_2, et M_3 et requises par M_3 et M_1. Il est ainsi possible de définir les matrices M_i^P suivantes :

$$\mathbf{M_1^P} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{M_2^P} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{M_3^P} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \mathbf{M_1^R} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{M_2^R} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{M_3^R} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Par application du modèle de composition, on réduit $M' = \{M_1,\ M_2,\ M_3,\ M_1M_2,\ M_1M_3,\ M_2M_3,\ M_1M_2M_3\}$ l'ensemble des compositions possibles, en $M = \{M_2,\ M_1M_2, M_1M_2M_3\}$ l'ensemble des compositions valides compte tenu des valeurs ci-dessus.

1. Application Data Unit

4.3. *Efficacité*

A partir des descriptions fonctionnelles précédentes, les matrices d'efficacité M_i^E pour les modules considérés dans les contextes IEEE 802.11 non chargé $(M_i^E)_1$, IEEE 802.11 chargé $(M_i^E)_2$, IEEE 802.11e non chargé $(M_i^E)_3$ et IEEE 802.11e chargé $(M_i^E)_4$ peuvent être établies. L'évaluation d'un module protocolaire pour un certain contexte est un problème à part entière que nous n'abordons pas ici ; cependant nous faisons l'hypothèse que cette évaluation a conduit aux valeurs suivantes :

$$\mathbf{M_i^E} = \begin{pmatrix} 802.11\,nc \\ 802.11\,c \\ 802.11e\,nc \\ 802.11e\,c \end{pmatrix} \quad \mathbf{M_1^E} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix} \quad \mathbf{M_2^E} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix} \quad \mathbf{M_3^E} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 2 \end{pmatrix}$$

A des fins d'illustration, les valeurs données découlent du raisonnement suivant : le contrôle de congestion et l'optimisation de la QdS sont plus efficaces dans le contexte d'un réseau congestionné que d'un réseau non chargé. De plus, le module propre à IEEE 802.11e n'a aucune influence hors du contexte pour lequel il a été conçu. Notons que ce type de comparaison va dans le sens d'études menées dans d'autres contextes réseaux [CEN 03].

4.4. *Expérimentation*

Pour confronter les résultats du modèle au contexte réseau ciblé (IEEE 802.11e), les compositions valides (M) ont été instanciées sur la plate-forme de simulation présentée figure 2. Leurs performances ont été évaluées pour le transfert d'un flux multimédia encodé en H.263 dont le profil de trafic est donné figure 2.

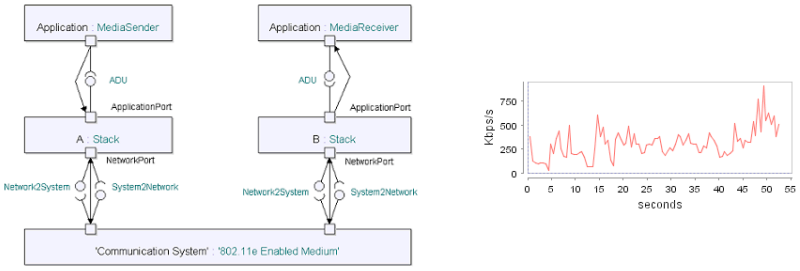


Figure 2. *Architecture considérée pour les expérimentations et profil de trafic de la vidéo*

L'évaluation des performances de ces compositions a été réalisée à l'aide d'une métrique applicative utilisée pour le codage vidéo : le *PSNR*.

Peak Signal Noise Ratio

Le PSNR permet de mesurer la qualité de présentation d'un flux vidéo transmis (potentiellement dégradé par rapport au flux émis). Le PSNR est défini comme suit.

Soient A et B les matrices de luminance de deux images vidéo (de dimensions $p \times q$) correspondant au même instant, respectivement sur la vidéo initiale (A) et sur la vidéo reçue (B). Étant donnée Y_{max} la valeur maximale de la luminance (dépendante de la représentation binaire choisie, généralement 255), la formule du PSNR est donnée par l'équation [3].

$$PSNR = 10log_{10}\left(\frac{Y_{max}^2}{\frac{1}{pq}\sum_{i=0}^{p-1}\sum_{j=0}^{q-1}(A_{ij}-B_{ij})^2}\right) \quad [3]$$

Ainsi, plus A et B sont proches (autrement dit, moins B est dégradée), plus la valeur du PSNR est élevée. Une valeur infinie traduit la restitution parfaite de l'image.

Modèle de réseau IEEE 802.11e
La simulation du réseau IEEE 802.11e est basée sur l'analyse des performances réalisée dans [HUA 05]. Dans cette étude, un modèle basé sur les chaînes de markov est utilisé pour décrire le mode EDCA (Enhanced DCF Channel Access) de IEEE 802.11e.

Cette étude montre l'existence d'un débit maximal pour chaque catégorie d'accès lorsque le système évolue à saturation (i.e. avec au moins un paquet à émettre pour chaque catégorie d'accès à tout instant). Dans tous les scénarii suivants, le réseau est considéré comme étant saturé par de nombreux flux dans les catégories « vidéo » et « best effort », de sorte qu'un nouveau flux bénéficie d'un débit maximum de 150Kbps pour la catégorie « best effort » et de 300Kbps pour la catégorie « vidéo ».

4.4.1. *Scenarii considérés*

Les différents scénarii sont synthétisés par la figure 3. Le but est d'évaluer chacune des compositions possibles et de confronter les résultats obtenus à la solution obtenue analytiquement au problème [2].

4.5. *Résultats*

4.5.1. *Résultats obtenus par le biais du modèle*

Si l'on considère le problème de décision défini par [2], toutes les compositions de M respectent les deux premières contraintes (qui sont les contraintes de validité [1] ayant servi pour la réduction de M'). Nous cherchons donc à évaluer l'efficacité pour chaque élément de M. Il vient $E(M_2)_4 = 2$, $E(M_2M_3)_4 = 4$, $E(M_1M_2M_3)_4 = 6$. Étant donné qu'il n'existe pas deux compositions à efficacité égale, nous ne cherchons pas à minimiser le coût des publications selon la troisième contrainte de [2].

Il ressort ainsi que la composition $M_1M_2M_3$ est la plus adaptée au contexte.

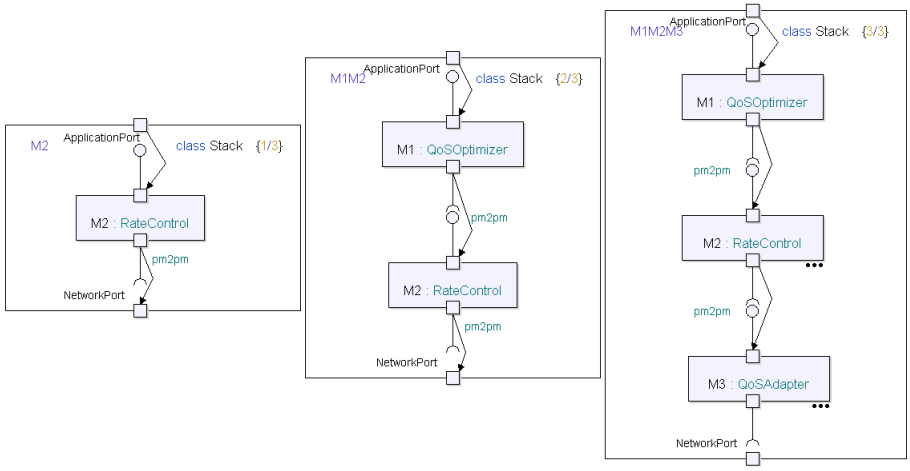


Figure 3. *Scenarii de test : Composition de l'élément* Stack

4.5.2. *Résultats obtenus par simulation*

Les résultats obtenus pour les trois scénarii sont présentés dans le tableau 1.

Pour chaque transmission, la part des images dont le calcul du PSNR retourne une valeur infinie (i.e. images non altérées par la transmission) est donnée conjointement à la valeur moyenne du PSNR. Cette valeur est calculée sur les images pour lesquelles le PSNR est fini. Elle rend compte, dans le cas où la présentation n'est pas parfaite, de la qualité moyenne de celle ci.

Composition	PSNR moyen (dB)	Images parfaites (%)
M_2	13,39	12
M_1M_2	15,51	14
$M_1M_2M_3$	23,61	44

Tableau 1. *Synthèse des résultats de simulation pour chaque composition*

De cette étude, il est possible de voir que la composition comprenant uniquement M_2 fournit les moins bonnes performances, aussi bien en termes d'images parfaites (12%) que pour le PSNR moyen. La composition M_1M_2 présente des performances intermédiaires, en effet, davantage d'images sont acheminées sans subir de distorsion (14%) et la présentation des autres images (dégradées) est légèrement améliorée (15,51dB). Enfin, lorsque l'on évalue la composition $M_1M_2M_3$, il apparaît que le nombre d'images parfaitement reçues est bien supérieur aux deux cas précédents (44%), et que les images dégradées présentent, globalement, une nette amélioration de qualité avec un PSNR moyen de l'ordre de 23,61dB.

De ces résultats, il apparaît donc que la réponse fournie au problème de décision [2], $M_1M_2M_3$ correspond au meilleur choix possible parmi les compositions de M.

5. Conclusion & Perspectives

Dans cet article, nous avons présenté un modèle analytique permettant de guider le processus de choix de composition d'architecture interne des protocoles dynamiquement configurables pour l'adaptation au contexte. Ce modèle a été appliqué à un contexte de réseau sans fil IEEE 802.11e, et confronté à des résultats de simulations menées pour l'ensemble des compositions de modules protocolaires candidates.

Les résultats obtenus ont montré que le modèle conduisait à la production de compositions valides dont le classement s'est avéré pertinent au regard des résultats de simulation.

Le modèle est paramétré par des critères d'efficacité des modules protocolaires, dont la valeur est essentielle pour garantir la pertinence de la décision. L'obtention de ces critères fait actuellement l'objet de nombreux travaux de recherche dans le domaine des protocoles (i.e. améliorations de TCP, *benchmarks* de DCCP, ...). L'un des points délicats concerne la caractérisation et la prise en compte de l'influence (positive ou négative) que peut avoir un module sur un autre en termes de performances.

Les perspectives directes de nos travaux visent à enrichir le modèle pour prendre en compte ces dépendances, non considérées jusque là. Par ailleurs, nous intégrerons également la prise en compte des besoins applicatifs dans l'optique de cibler une adaptation à un contexte dynamique à deux niveaux (besoins applicatifs d'une part et environnement réseau d'autre part).

6. Remerciements

Ces travaux sont actuellement menés dans le cadre du projet européen NetQoS (http ://www.netqos.eu)

7. Bibliographie

[AKA 04] AKAN O. B., AKYILDIZ. I. F., « ATL : An Adaptive Transport Layer Suite for Next-Generation Wireless Internet. », *IEEE Journal on Selected Areas in Communications*, vol. Vol. 22, n° No. 5, June 2004, p. 802-817.

[AME 94] AMER P., CHASSOT C., CONNOLLY C., CONRAD P., DIAZ M., « Partial Order Transport Service for Multimedia and other Applications », *IEEE ACM Transactions on Networking*, vol. vol.2, no. 5, October 1994.

[BAL 04] BALASUBRAMANIAM S., INDULSKA J., « Vertical handover supporting pervasive computing », *Elsevier Computer Communication Journal, Special Issue on 4G/Future Wireless networks*, vol. vol. 27/8, 2004, p. 708-719.

[BHA 98] BHATTI N. T., HILTUNEN M. A., SCHLICHTING R. D., CHIU W., « Coyote : a system for constructing fine-grain configurable communication services », *ACM Transactions on Computer Systems*, vol. 16, n° 4, 1998, p. 321–366.

[BHA 99] BHATTI N., SCHLICHTING R., « Configurable Communication Protocols for Mobile Computing », *The Fourth International Symposium on Autonomous Decentralized Systems*, 1999, p. 220-227.

[BRI 01] BRIDGES P. G., CHEN W.-K., HILTUNEN M. A., SCHLICHTING R. D., « Supporting Coordinated Adaptation in Networked Systems », *8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Elmau, Germany*, May 2001.

[CEN 03] CEN S., COSMAN P., VOELKER G., « End-to-end differentiation of congestion and wireless Losses », *ACM Multimedia Computing and Networking*, 2003.

[CHA 06] CHASSOT C., GUENNOUN K., DRIRA K., ARMANDO F., EXPOSITO E., LOZES A., « Towards Autonomous Management of QoS through Model-Driven Adaptability in Communication-Centric Systems », *International Transactions on Systems Science and Applications*, vol. 2, n° 3, 2006, p. 255-264.

[CHE 05] CHEN L., W.B. H., « QoS-aware routing based on bandwidth estimation for mobile ad hoc network », *IEEE Journal on Selected Areas in Communications*, vol. Volume 23, Issue 3, March 2005, p. 561-572.

[DAS 04] DASILVA L. A., MIDKIFF S. F., PARK J. S., HADJICHRISTOFI G. C., DAVIS N. J., PHANSE K. S., LIN T., « Network Mobility and Protocol Interoperability in Ad Hoc Networks », *IEEE Communications Magazine*, vol. Volume 42, Issue 11, November 2004, p. 88-96.

[EXP 03a] EXPOSITO E., DIAZ M., SENAC P., « FPTP : the XQoS aware and fully programmable transport protocol », *11th IEEE International Conference on Networks (ICON'2003), Sydney (Australie) - pp.249-254*, 28 September - 1st October 2003.

[EXP 03b] EXPOSITO E. J., « Spécification et Mise en oeuvre d'un protocole de Transport orienté Qualité de Service pour les applications Multimédias », PhD thesis, Institut National Polytechnique de Toulouse, spécialité Réseaux et Télécommunications, 2003.

[FAR 06] FARKAS K., WELLNITZ O., DICK M., GU X., BUSSE M., EFFELSBERG W., REBAHI Y., SISALEM D., GRIGORAS D., STEFANIDIS K., SERPANOS D., « Real-time service provisioning for mobile and wireless networks », *Elsevier Computer Communication Journal*, vol. Vol. 29, No. 5, march 2006, p. 540-550.

[FLO 04] FLOYD S., KOHLER E., « Profile for DCCP Congestion Control ID 3 : TFRC Congestion Control », rapport, December 2004, Internet Draft.

[HAN 03] HANDLEY M., PAHDYE J., FLOYD S., , WIDMER J., « TCP Friendly Rate Control (TFRC) : Protocol Specification », Internet draft draft-ietf-tsvwg-tfrc02. txt, work in progress, January 2003.

[HUA 05] HUA Z., CHLAMTAC I., « Performance analysis for IEEE 802.11e EDCF service differentiation », *IEEE Transactions on Wireless Communications*, vol. 8, 2005, p. 1779-1788.

[HUT 91] HUTCHINSON N. C., PETERSON L. L., « The x-Kernel : An Architecture for Implementing Network Protocols », *IEEE Transactions on Software Engineering*, vol. 17, n° 1, 1991, p. 64–76, IEEE Computer Society.

[KAL 06] KALOXYLOS A., LAMPROPOULOS G., PASSAS N., MERAKOS L., « A flexible handover mechanism for seamless service continuity in heterogeneous environments », *El-*

sevier Computer Communications, vol. Volume 29, Issue 6, 31 March 2006, p. 717-729.

[KON 02] KON F., COSTA F., BLAIR G., CAMPBELL R. H., « The case for reflective middleware », *Commun. ACM*, vol. 45, n° 6, 2002, p. 33–38, ACM Press.

[LAN 04] LANDRY R., GRACE K., SAIDI A., « On the Design and Management of Heterogeneous Networks : A Predictability-Based Perspective », *IEEE Communications Magazine*, vol. Volume 42, Issue 11, November 2004, p. Page(s) :80 - 87.

[MEN 03] MENA S., CUVELLIER X., GRÉGOIRE C., SCHIPER A., « Appia vs. Cactus : Comparing Protocol Composition Frameworks », *Proceeidngs of the 22nd Symposium on Reliable Distributed Systems (2003)*, 2003.

[MIR 01] MIRANDA H., PINTO A., RODRIGUES L., « Appia : A Flexible Protocol Kernel Supporting Multiple Coordinated Channels », *Proc. 21st International conference on Distributed Computing Systems (ICDCS-21), (Phoenix, Arizona, USA)*, 2001, p. 707–710.

[MOC 05] MOCITO J., ROSA L., ALMEIDA N., MIRANDA H., RODRIGUES L., LOPES A., « Context Adaptation of the Communication Stack », *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)*, Colombus, OH, USA, June 2005, Colombus.

[PER 04] PEREZ G., SKARMETA A. G., « Policy-Based Dynamic Provision of IP Services in a Secure VPN Coalition Scenario », *IEEE Communications Magazine*, vol. Volume 42, Issue 11 :, November 2004, p. 118 - 124.

[WON 01] WONG G. T., HILTUNEN M. A., SCHLICHTING R. D., « A Configurable and Extensible Transport Protocol », *Proceedings of INFOCOM*, Anchorage, Alaska, USA, April 2001, p. 319-328.

[WU 01] WU D., HOU Y. T., ZHU W., « Streaming Video over the Internet : Approaches and Directions », *IEEE Transactions on Circuits and Systems for Video Technology*, vol. vol. 11, no. 1, February 2001.

[WU 05] WU R., CHIEN A., HILTUNEN M., SCHLICHTING R., SEN S., « A High Performance Configurable Transport Protocol for Grid Computing », *Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), Cardiff, Wales*, 2005.

[YU 03] YU F., ZHANG Q., ZHU W., ZHANG Y.-Q., « QoS-adaptive proxy caching for multimedia streaming over the Internet », *IEEE Transactions on Circuits and Systems for Video Technology*, vol. Volume 13, Issue 3, March 2003, p. 257 - 269.

HERMÈS SCIENCE PUBLICATIONS
8 Quai du Marché Neuf - 75004 Paris
Tel : 01-53-10-15-20 Télécopie : 01-53-10-15-21
Email : hermes@hermes-science.com
Serveur Web : http ://www.hermes-science.com

Annexe pour le service de fabrication

Article pour les actes :
CFIP 08

Auteurs :
Nicolas Van Wambeke — François Armando — Christophe Chassot — Ernesto Expósito

Titre de l'article :
Un modèle de décision pour l'auto-adaptation des protocoles de communication

Titre abrégé :
Protocoles auto-configurables

Traduction du titre :
A decision model for communication protocol's self-adaptation

Date de cette version :
8 février 2008

Coordonnées des auteurs :

– téléphone : 05.61.33.69.07
– télécopie : 05.61.33.64.11
– Email : van.wambeke@laas.fr

Logiciel utilisé pour la préparation de cet article :
LaTeX, avec le fichier de style `article-hermes.cls`,
version 1.10 du 17/09/2001.

Formulaire de copyright :
Joindre le formulaire de copyright signé, récupéré sur le web à l'adresse `http://www.hermes-science.com`

Internet

Explication et réduction de l'impact des violations d'inégalités triangulaires dans Vivaldi

François Cantin* — **Bamba Gueye*** — **Mohamed Ali Kaafar*** — **Guy Leduc*** — **Laurent Mathy****

* *Université de Liège*
Institut Montefiore B 28, B-4000, Liège Belgique
{francois.cantin, cabgueye, ma.kaafar, guy.leduc}@ulg.ac.be

** *University of Lancaster*
Lancaster, UK LA1 4WA
laurent@comp.lancs.ac.uk

RÉSUMÉ. *Les systèmes de coordonnées sont des systèmes distribués ayant pour but, à partir de mesures de distance (par exemple RTT) entre certaines paires de nœuds, d'associer des coordonnées à chaque nœud dans un espace métrique. Toutefois, de tels systèmes ne fonctionnent pas correctement lorsque les distances mesurées ne respectent pas les inégalités triangulaires. Or, les violations de ces inégalités, appelées TIV, sont fréquentes dans l'Internet. Nous proposons une étude approfondie de l'impact des TIV sur le système de coordonnées Vivaldi. Nous quantifions et expliquons les erreurs de prédiction de distance et l'instabilité des coordonnées causées par les TIV selon leur fréquence et leur sévérité. Nous montrons aussi que la distance entre deux nœuds, mesurée par le RTT, est corrélée à la probabilité d'existence d'une TIV. Enfin, nous recommandons un système de coordonnées hiérarchique, et nous montrons, par des simulations sur une matrice de délais réelle, qu'une telle approche réduit l'impact des TIV.*

ABSTRACT. *Network coordinate systems embed delay measurements (e.g. RTT) between Internet nodes into some metric space. These systems often assume the triangle inequality holds for Internet delays. However, the reality is that the triangle inequality is violated by Internet delays. In a first step, we explore the ways in which TIVs impact the Vivaldi coordinate system using different metrics in order to quantify various levels of TIV's severity. In a second step, we study TIVs existing in the Internet. Our results show that path lengths do have an effect on the impact of these TIVs. In particular, we observed correlation between the (in)stability and high effective error of nodes' coordinates with respect to their involvement in TIVs situations. Finally, we propose a Two-Tier architecture that does mitigate the effect of TIVs on the distance predictions.*

MOTS-CLÉS : *Systèmes de coordonnées, Performance, Violation de l'inégalité triangulaire*

KEYWORDS: *Internet Coordinate Systems, Performance, Triangle Inequality Violations*

1. Introduction

Nous avons assisté ces dernières années à une énorme croissance d'applications Internet [ROW 01, KUB 00, CHU 00, SKY], basées et/ou bénéficiant des réseaux de recouvrement (ou overlay). Ces applications tiennent compte de la topologie du réseau physique pour la construction du réseau de recouvrement. En particulier, la plupart de ces applications (si ce n'est toutes) et leur réseau de recouvrement associé, se basent sur la notion de proximité réseau, typiquement définie en termes de délais aller-retour (RTT), pour l'optimisation de la sélection de voisins. Cependant, les mesures de proximité peuvent s'avérer extrêmement coûteuses en termes de consommation en bande passante. En effet, l'existence simultanée de plusieurs réseaux de recouvrement, peut entraîner un surcoût de communications élevé, dû aux mesures de proximité individuelles menées par chaque nœud de ces réseaux.

De plus, traquer la proximité au sein d'un groupe dynamique, nécessite une fréquence de mesure très grande. Cela induit encore un surcoût de mesures plus important. Afin de pallier ce problème, des systèmes de positionnement "Internet", comme [NG 02, PIA 03, COS 04, SHA 03, DAB 04], ont été introduits. Dans ces systèmes, l'idée principale est que si chaque nœud peut être associé à des coordonnées virtuelles dans un espace approprié, la distance entre les nœuds est trivialement calculée sans pour autant avoir recours à des mesures directes. En d'autres termes, ces systèmes plongent les mesures des temps de latence (délais) entre nœuds dans un espace métrique et associent un vecteur de coordonnées (ou coordonnées) dans cet espace à chaque nœud, dans le but de permettre des prédictions de distances précises et peu onéreuses pour n'importe quelle paire de nœuds dans le réseau. L'avantage premier de ces systèmes est que si les distances réseaux (dans le sens de délais) sont plongées dans un espace de coordonnées où une position raisonnablement précise pour chaque nœud est établie, le surcoût de mesures produit par le positionnement, est amorti sur plusieurs prédictions de distances. Ceci réduit énormément le coût en termes de mesures de distances du système entier.

Cependant, les politiques de routages et l'existence de congestion sur certain chemins, peuvent entraîner des *violations du principe de l'inégalité triangulaire* ("Triangle Inequality Violation" - TIV) sur les délais dans l'Internet [ZHE 05] . Ces violations seraient la cause de distorsions et d'erreurs de prédiction pour les systèmes de coordonnées. Prenons l'exemple de trois nœuds A, B et C tels que $d(A, B)$ est de $1\ ms$, $d(B, C)$ est de $2\ ms$ et $d(A, C)$ est de $5\ ms$ où $d(X, Y)$ dénote le délai existant entre le nœud X et le nœud Y. Dans ce cas, le principe de l'inégalité triangulaire est violé car $d(A, C) > d(A, B) + d(B, C)$. Ainsi, l'immersion (ou "*embedding*") exacte des délais dans un espace métrique, où l'inégalité triangulaire doit être respectée, est forcément impossible. Face à des situations de TIV, les nœuds d'un système de coordonnées auront tendance à alterner entre des sous-estimations et des sur-estimations de la distance réelle, sans jamais parvenir à se positionner dans l'espace métrique d'une manière parfaite.

Exploiter les systèmes de coordonnées pour diverses opérations de prédiction de distances au niveau applicatif nécessite en revanche, que les coordonnées de ces systèmes soient aussi précises que stables.

Afin d'améliorer la précision des systèmes de coordonnées, certains travaux ont envisagé l'exclusion des nœuds qui forment des TIV entre eux [LED 07, Y.B 03]. Toutefois, sacrifier ne fusse qu'une infime partie des nœuds ne nous semble pas justifiable car les TIV sont une propriété naturelle inhérente à l'Internet. Les auteurs de [WAN 07] proposent d'éliminer parmi l'ensemble des voisins d'un nœud A tous les nœuds B_i telle que la distance $d(A, B_i)$ prédite à partir des coordonnées est fortement sous-estimée. Ainsi, la majorité des voisins de A sont choisis parmi les nœuds qui lui sont très proches. Ceci entraîne une sélection non hybride (*i.e.* proches et éloignés) des voisins, alors qu'une telle propriété est une condition nécessaire pour le bon fonctionnement des systèmes de coordonnées.

Dans cet article, nous étudions d'abord l'impact des TIV sur le système de coordonnées Vivaldi. Cette étude montre que les nœuds les plus impliqués dans les TIV sont en général deux fois moins précis comparés aux autres nœuds, quant à leur positionnement dans le système de coordonnées. Leurs coordonnées sont également moins stables, avec de fortes amplitudes d'oscillations. Dans une deuxième étape, nous analysons la distribution des TIV dans l'Internet et par la suite, caractérisons leur sévérité en considérant différentes métriques. Nous montrons que les chemins les plus longs sont ceux qui sont les plus impliqués dans des TIV sévères. Ainsi, nous proposons une structure hiérarchique, opposée à la structure plate de Vivaldi, se basant sur un regroupement de nœuds proches. Les nœuds appartenant à un même groupe calculent des coordonnées locales pour prédire les distances entre eux. En revanche, au niveau supérieur de la hiérarchie, nous considérons un Vivaldi "plat" afin de prédire les distances entre les nœuds de différents groupes. Cette approche hiérarchique limite l'impact des TIV sur les estimations de distances, et permet une meilleure prédiction pour les petits chemins.

2. Vivaldi et les violations de l'inégalité triangulaire

2.1. *Description de Vivaldi*

Vivaldi [DAB 04] est un système de coordonnées complètement décentralisé. Il est basé sur une simulation de ressorts, où la position du nœud correspond à celle de l'extrémité d'un ressort qui minimiserait l'énergie potentielle des ressorts, et donc minimiserait l'erreur de positionnement. Un nœud se joignant au système, calcule ses coordonnées en collectant des informations de positions et de mesures de délai à partir de quelques autres nœuds. Spécifiquement, Vivaldi place un ressort entre chaque paire de nœuds (i, j) dans le système, avec une longueur "au repos" correspondant à la mesure RTT entre ces deux nœuds. La longueur réelle du ressort est considérée comme étant l'estimation de distance entre les deux positions des nœuds. L'énergie potentielle d'un tel ressort est proportionnelle au carré du déplacement par rapport à sa longueur

au repos. La somme de ces erreurs à travers tous les ressorts est la fonction d'erreur que Vivaldi tente de minimiser. Une procédure identique tourne sur tous les nœuds Vivaldi. Celle-ci est basée sur des échantillons récoltés par les nœuds qui fournissent les informations pour leur positionnement. Un échantillon, utilisé par un nœud i est ainsi constitué de la mesure vers un nœud j, de la coordonnée du même nœud j, ainsi que de l'erreur locale reportée par j. L'algorithme traite les nœuds à haute erreur, en assignant des poids à chaque échantillon collecté. L'erreur relative de cet échantillon, e_e, est alors calculée comme suit :

$$e_e = \frac{|\parallel x_j - x_i \parallel - RTT(i,j)_{mesure}|}{RTT(i,j)_{mesure}}$$

où x_i et x_j représentent respectivement les coordonnées des nœuds i et j. Le nœud i met à jour son erreur locale comme suit : $e_i = e_e \times w_e + e_i \times (1 - w_e)$ où $w_e = e_i/(e_i + e_j)$ est le poids de l'échantillon calculé à partir de son erreur locale e_i et de l'erreur locale e_j du nœud j. Le nœud utilise aussi ce poids pour calculer la valeur $\delta_e = C_c \times w_e$, avec $0 < C_c < 1$, qui définit l'amplitude du déplacement du nœud dans la direction spécifiée par cet échantillon. A l'aide de δ_e, le nœud i met à jour ses coordonnées de la manière suivante :

$$x_i = x_i + \delta_e \cdot (RTT(i,j)_{mesure} - \parallel x_i - x_j \parallel) \cdot u(x_i - x_j)$$

où $u(x_i - x_j)$ est un vecteur unitaire indiquant la direction dans laquelle le noeud i doit se déplacer.

2.2. *Critères de sévérité des TIV*

Des études antérieures tels que [SAV 99, ZHE 05, LEE 06] ont caractérisé les TIV dans l'Internet en considérant le ratio et le pourcentage de triangles victimes de TIV. Soit un triangle ABC tel que, par convention, AB représente le côté le plus long de ce triangle. Si $d(A,B) > d(A,C) + d(C,B)$, alors ABC forme une TIV car le principe de l'inégalité triangulaire est violé. Nous proposons deux critères de sévérité des TIV. Le premier critère est la *sévérité relative* et est définie comme suit :

$$G_r = \frac{d(A,B) - (d(A,C) + d(C,B))}{d(A,B)} \quad (1)$$

La valeur de G_r varie entre 0 (sévérité minimale) et 1 (sévérité maximale) et permet de caractériser le gain relatif obtenu en termes de distance en empruntant le chemin indirect. Toutefois, lorsque la distance $d(A,B)$ est petite, la sévérité relative peut être grande sans pour autant que la TIV soit réellement sévère. Nous définissons alors un second critère dénomé *sévérité absolue*, comme suit :

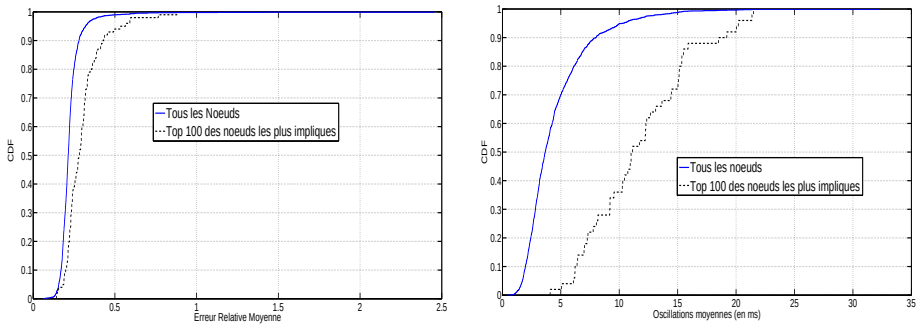
$$G_{ar} = \frac{d(A,B) - (d(A,C) + d(C,B))}{\mathcal{D}} \quad (2)$$

G_{ar} représente le gain absolu en termes de distance normalisée par le diamètre de l'espace dénoté $\mathcal{D}$. G_{ar} varie également entre 0 (sévérité minimale) et 1 (sévérité maximale). Par la suite, nous fixons différents seuils de sévérité th_r et th_{ar} et considérons toutes les TIV telles que $G_r \geq th_r$, ou $G_{ar} \geq th_{ar}$, ou bien celles supérieures aux deux seuils à la fois.

2.3. *Impact des TIV dans Vivaldi*

Pour nos simulations, nous avons utilisé le simulateur *p2psim* [P2P] qui fournit une implémentation de Vivaldi. Chaque nœud Vivaldi choisit 32 voisins et la moitié d'entre eux sont choisis parmi les nœuds les plus proches. La constante C_c (voir section 2.1) est fixée à 0,25 (valeur recommandée dans [DAB 04]). Le système est considéré comme "ayant convergé" ou "s'étant stabilisé" si toutes les erreurs relatives des nœuds convergent vers des valeurs qui varient au maximum de 0,02 pour 10 pas de simulation consécutifs. Toutes les simulations convergent en moins de 1800 pas (ce qui correspond à peu près à 8 heures, un pas représentant environ 17 secondes).

Pour nos simulations, nous avons considéré un espace euclidien à deux dimensions. Nous utilisons l'ensemble de données "King" [GUM 02] qui contient les RTT mesurés entre 1740 serveurs *DNS* [P2P]. Pour étudier l'impact des TIV sur la précision de la prédiction de Vivaldi, nous définissons le degré d'implication d'un nœud dans une TIV. Pour chaque nœud C, ce degré d'implication est le nombre de paires (A_i, B_j), $i \neq j$, existantes où A_iB_jC est un triangle violant l'inégalité triangulaire. Sur la base de ce degré d'implication, nous comparons la précision de la prédiction de distance des 100 nœuds les plus impliqués dans des TIV à celle de l'ensemble des nœuds du système.



(a) CDF de l'erreur relative moyenne (b) CDF des oscillations

Figure 1. *Impact des TIV sur la précision de la prédiction de Vivaldi.*

La figure 1(a) montre la CDF (*Cumulative Distribution Function*) de l'erreur relative moyenne des 100 nœuds les plus impliqués dans des TIV comparée à celle de l'ensemble des nœuds du système (S). L'*erreur relative moyenne* (ERM) pour un

nœud est calculée comme étant la moyenne des erreurs relatives de prédiction entre ce nœud et tous les autres nœuds du système lors des derniers pas de simulation.

$$ERM(i) = \frac{\sum_{j \neq i \in S} \frac{|(RTT(i,j) - ||x_i - x_j||)|}{RTT(i,j)}}{|S|}$$

La figure 1(a) montre que la prédiction de distance est moins précise pour les nœuds les plus impliqués dans des TIV. En effet, en considérant l'ensemble des nœuds, plus de 90 % d'entre eux ont une ERM inférieure à 0,3, alors que pour les 100 nœuds les plus impliqués dans des TIV le pourcentage de nœuds atteignant cette valeur est seulement de 50 %.

Nous avons également observé l'oscillation des coordonnées des nœuds dans le système Vivaldi. L'oscillation d'un nœud est définie comme étant la distance entre les coordonnées du nœud lors de deux pas successifs. Nous représentons l'oscillation moyenne d'un nœud comme étant la moyenne de ses oscillations lorsque l'on considère les 500 derniers pas de la simulation. La figure 1(b) montre la CDF des oscillations moyennes des 100 nœuds les plus impliqués dans des TIV comparée à celle de l'ensemble des nœuds du système. Nous remarquons une fois de plus que l'influence des TIV sur la prédiction de distance est assez importante puisque les nœuds les plus impliqués dans des TIV ont une oscillation moyenne beaucoup plus importante que celle des autres nœuds. 70 % des nœuds ont une oscillation inférieure à $5\ ms$ alors qu'uniquement 3 nœuds parmi les 100 les plus impliqués dans des TIV voient leur coordonnées osciller en moyenne de moins de $5\ ms$.

Nous venons de montrer que les TIV ont à la fois un impact considérable sur la précision du système de coordonnées Vivaldi, mais aussi sur la stabilité de ses coordonnées. Dans la section suivante, nous étudions la distribution des TIV existant dans l'Internet et caractérisons leur sévérité en utilisant les deux critères de sévérités présentés dans la section 2.2.

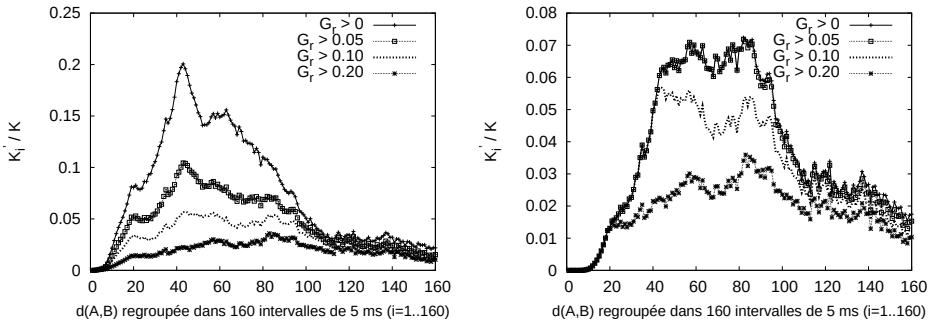
3. Analyse des violations de l'inégalité triangulaire dans l'Internet

Soit K le nombre total de triangles ABC obtenus en considérant l'ensemble des nœuds de la matrice de délai King. Nous obtenons K = 854 773 676 parmi lesquels 105 329 511 triangles sont victimes de TIV (12 % du nombre total de triangles). Les RTT mesurés entre les nœuds sont répartis dans 160 intervalles de $5\ ms$ chacun (les RTT variant de $5\ ms$ à $800\ ms$ dans la matrice de délai). Nous supposons que le triangle ABC est dans l'intervalle i ($0 \leq i < 160$), si d(A, B) (son côté le plus long) appartient à cet intervalle. Nous dénoterons par K_i le nombre de triangles appartenant à l'intervalle i et par K'_i le nombre de triangles qui forment des TIV à l'intérieur de l'intervalle i.

Soit P_i la probabilité qu'un triangle ABC choisi au hasard soit victime d'une TIV et appartienne à l'intervalle i. Cette probabilté est calculée comme suit :

$$P_i = \frac{K'_i}{K_i} \times \frac{K_i}{K} = \frac{K'_i}{K} \tag{3}$$

La figure 2 montre la distribution des P_i en considérant les deux critères de sévérité définis dans la section 2.2.



(a) th_{ar} = 0 (b) th_{ar} = 0,025

Figure 2. *Distribution des TIV pour différents seuils de sévérité.*

Sur la figure 2(a), nous ne considérons que la sévérité relative ($th_{ar} = 0$) alors que sur la figure 2(b), nous ne tenons pas compte des TIV dont la sévérité absolue est inférieure à th_{ar} = 0,025. La figure 2 montre qu'il est peu probable d'avoir une TIV sévère pour les triangles tels que $d(A, B) < 150\ ms$ ($i \leq 30$). Quel que soit le degré de sévérité des TIV, les triangles tels que $d(A, B) > 200\ ms$ ($i \geq 40$) ont plus de chance d'être des TIV sévères. Au vu de ces observations nous proposons une approche hiérarchique pour limiter l'impact des TIV. Ainsi, en définissant des groupes dont le diamètre est inférieur à $150\ ms$, il est peu probable que les TIV à l'intérieur des groupes soient sévères. Les coordonnées calculées à l'intérieur d'un groupe seraient dès lors plus précises de même que la prédiction des chemins à l'intérieur d'un groupe.

4. Vivaldi hiérarchique

Cette section est constituée de trois parties. Dans un premier temps, nous donnons un aperçu de l'approche hiérarchique de Vivaldi. Ensuite, nous décrivons l'algorithme de regroupement et les groupes utilisés lors de notre étude. Enfin, nous comparons les résultats de notre approche hiérarchique à ceux obtenus avec le Vivaldi "plat".

4.1. *Aperçu*

Nous proposons un Vivaldi hiérarchique dans lequel les nœuds sont regroupés en ensemble de nœuds proches. Chaque nœud choisit 32 voisins dont 16 sont à l'inté-

rieur de son cluster (voisins proches) et 16 autres en dehors. Les nœuds se trouvant dans chaque groupe calculent deux vecteurs de coordonnées : des coordonnées locales et des coordonnées globales. Les coordonnées locales sont mises à jour en réalisant des mesures avec les 16 voisins membres du même groupe. Elles sont utilisées uniquement pour la prédiction des chemins intra-groupe. Pour la mise à jour des coordonnées globales, chaque nœud considère ses 32 voisins. Celles-là sont utilisées pour la prédiction des chemins inter-groupe. Nous décrivons par la suite comment les clusters considérés dans notre étude ont été construits.

4.2. *Le processus de regroupement*

Dans un premier temps nous avons basé la construction des groupes sur les coordonnées fournies par un Vivaldi “plat” dans un espace euclidien à deux dimensions. Ces coordonnées $2D$, obtenues en appliquant Vivaldi sur la matrice de délais, permettent d’observer 5 principaux nuages de points (groupes) distincts. Pour cette étude, nous avons sélectionné les trois groupes les plus peuplés en fixant un diamètre maximum. Celui ci variant de $60\ ms$ à $140\ ms$ pour les 3 groupes. Les nœuds qui ne font pas partie d’un des 3 groupes participent uniquement au niveau supérieur du Vivaldi hiérarchique et ne calculeront donc que des coordonnées globales.

Toutefois, il est important de noter que les groupes tels que définis à ce niveau sont basés sur des estimations de RTT faites grâce aux coordonnées fournies par un Vivaldi “plat”. Ainsi, il peut arriver que certains nœuds soient mal placés, et par conséquent engendrent un diamètre beaucoup plus grand que prévu pour les groupes. Dans une seconde étape, nous vérifions alors dans chaque groupe si les nœuds respectent les contraintes de délai fixées en tenant compte des mesures de RTT réelles. Pour chaque groupe, nous appliquons de manière récursive l’algorithme ci-dessous jusqu’à ce qu’aucune paire de nœuds appartenant au même groupe ne viole le diamètre maximum fixé :
1. Pour chaque nœud, nous vérifions si le RTT avec les autres nœuds du groupe ne dépasse pas le diamètre fixé ; chaque dépassement entraîne l’incrémentation d’un compteur pour ce nœud.
2. Nous retirons ensuite du groupe le nœud qui compte le plus de violations[1].

Le tableau 1 montre les caratéristiques des groupes obtenus.

Dans ce qui suit, nous comparons les performances obtenues en utilisant le Vivaldi hiérarchique et le Vivaldi “plat” et ce vis-à-vis de la précision de positionnement des nœuds et de l’amplitude moyenne des oscillations de leurs coordonnées.

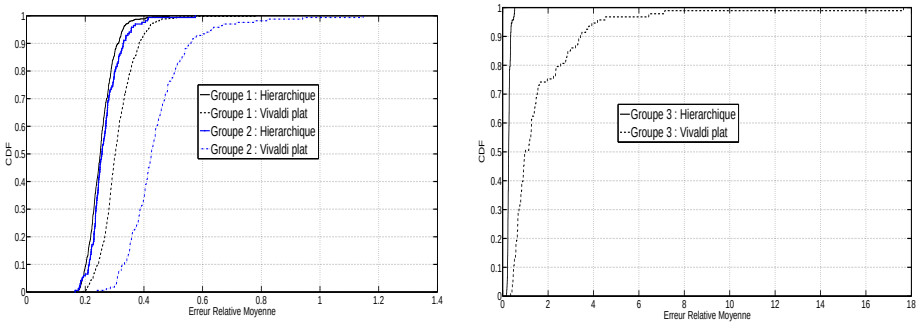
1. Ce nœud a très probablement une grande erreur relative occasionnant une erreur de positionnement importante.

	Nombre de nœuds	Diamètre
Groupe 1	565	140 ms
Groupe 2	169	100 ms
Groupe 3	93	60 ms

Tableau 1. *Caractéristiques des groupes*

4.3. *Evaluation des performances du Vivaldi hiérarchique*

Afin de caractériser la précision de positionnement des nœuds, nous utilisons l'erreur relative moyenne des nœuds (ERM - voir section 2.3) comme indicateur de performance. Les figures 3(a) et 3(b) montrent les CDF des erreurs relatives moyennes pour les différents groupes. Pour un groupe donné, seuls les nœuds de ce groupe sont considérés et les ERM sont calculées uniquement sur la base des erreurs de prédiction des chemins intra-groupe. Pour chaque groupe, nous avons une courbe représentant le Vivaldi hiérarchique, où les coordonnées locales des nœuds sont utilisées pour estimer les erreurs de prédiction, et une représentant le Vivaldi plat en considérant le même ensemble de nœuds.



(a) groupes 1 et 2 (b) groupe 3

Figure 3. *Comparaison des erreurs relatives moyennes.*

Nous remarquons clairement que les erreurs relatives moyennes calculées à l'aide des coordonnées locales sont plus faibles que celles calculées à l'aide des coordonnées du Vivaldi "plat". Par exemple, dans le groupe 2, plus de 90 % des nœuds ont une ERM inférieure à 0,3 en considérant les coordonnées locales alors qu'en considérant les coordonnées du Vivaldi plat, un peu moins de la moitié des nœuds du groupe ont une ERM supérieure à 0,5. C'est dans le groupe 3 que nous observons la plus grande différence entre le Vivaldi hiérarchique et le Vivaldi plat. En effet, sur la figure 3(b), la quasi totalité des nœuds du groupe ont une ERM inférieure à 0,5 en considérant les coordonnées locales. Par contre, 70 % de ces nœuds ont une très mauvaise erreur relative moyenne lorsque les coordonnées du Vivaldi "plat" sont utilisées.

Ces résultats montrent que la prédiction de distance faite à l'intérieur des groupes est nettement meilleure en considérant le Vivaldi hiérarchique. Nous pouvons en déduire que plus le diamètre du groupe est petit, plus la différence de précision entre le Vivaldi hiérarchique et le Vivaldi plat est importante. La longueur des chemins est corrélée à la sévérité des TIV.

La meilleure précision de positionnement obtenue à l'intérieur des groupes est due au fait que pour calculer leurs coordonnées locales, les nœuds choisissent tous leurs voisins à l'intérieur de leur groupe. Ceci a pour effet de limiter au diamètre du groupe le délai maximal mesurable entre un nœud et son voisin. Ainsi, la valeur $G_a = d(A,B) - (d(A,C) + d(C,B)$ est limitée au diamètre du groupe. Si on considère le Vivaldi plat, cette valeur varie jusqu'à $800\ ms$. Des TIV ayant une valeur de G_a assez grande risquent d'introduire des erreurs d'estimation absolues importantes. Ainsi, lors de la mise à jour de leurs coordonnées, des nœuds risquent d'être confrontés à de grandes valeurs pour $RTT(i,j)_{mesure} - \parallel x_i - x_j \parallel$ (*voir* section 2.1). De tels nœuds oscillent sur de plus grandes amplitudes qui engendrent une imprécision de l'estimation de distance. Par contre, si les TIV ont toutes des petits G_a, alors les nœuds oscillent sur de petits intervalles permettant ainsi une meilleure prédiction des délais.

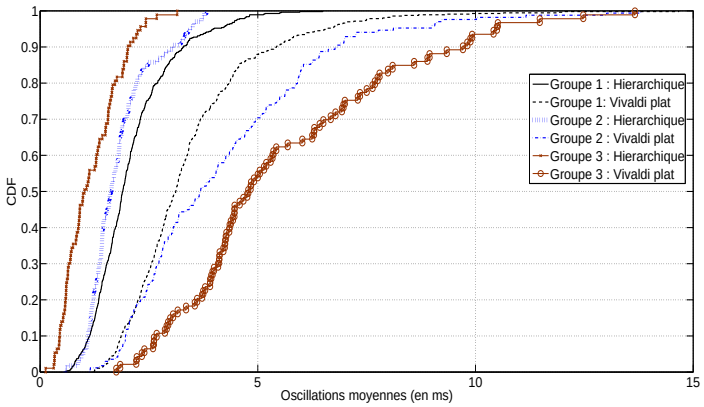


Figure 4. *Comparaison des oscillations des coordonnées pour le Vivaldi "plat" et le hiérarchique.*

Nous avons également observé les oscillations moyennes des coordonnées durant les 500 derniers pas de simulation. Les 3 CDF les plus à gauche de la figure 4 représentent les oscillations des coordonnées locales calculées dans nos 3 groupes. Ces résultats montrent clairement que les coordonnées locales oscillent avec une amplitude plus faible comparées aux coordonnées du Vivaldi plat. En effet, en considérant les coordonnées locales, plus de 80 % des nœuds ont une oscillation moyenne inférieure à $3\ ms$ alors que seuls 40 % des nœuds sont concernés si on considère les coordonnées du Vivaldi plat.

5. Conclusion

Nous avons proposé une approche hiérarchique qui permet d'atténuer l'impact des violations des inégalités triangulaires dans l'Internet. Jusqu'ici, les études sur les systèmes de coordonnées étaient généralement concentrées sur la recherche d'espaces ayant des propriétés géométriques particulières permettant un meilleur positionnement des nœuds. Notre proposition est plutôt basée sur un regroupement des nœuds proches de manière à atténuer l'impact des TIV. Les nœuds calculent des coordonnées locales plus précises pour estimer les chemins intra-groupes, et ils conservent des coordonnées globales pour estimer des chemins inter-groupes.

Même si cette étude s'est focalisée sur le système Vivaldi, l'architecture proposée peut être utilisée avec n'importe quel système de coordonnées. En effet, il suffit de faire tourner une instance du système de coordonnées dans chaque groupe et une instance au niveau global sans rien changer au protocole utilisé.

Lors de l'analyse de l'impact des TIV sur les estimations de distances, nous avons considéré qu'un nœud est impliqué dans une TIV s'il fait partie d'un triangle qui est une TIV (en tant que nœud A, B ou C). Toutefois, dans certaines situations, il se peut qu'un nœud ne soit pas (ou moins) affecté par les TIV existant avec les nœuds qu'il n'a pas choisi comme voisins. D'autres définitions de la notion de degré d'implication dans des TIV pourraient donc être envisagées de manière à analyser plus en détail l'impact des TIV sur la prédiction de distance. Par exemple, nous pourrions considérer qu'un nœud est impliqué dans une TIV uniquement s'il fait partie du TIV *et* s'il a choisi les deux autres membres du TIV comme voisins. Si nous arrivions à mettre en évidence un lien entre la relation de voisinage, les TIV et la précision de la prédiction, nous pourrions utiliser ces résultats pour choisir les voisins de manière à atténuer autant que possible l'impact des TIV au niveau supérieur de notre architecture.

La technique de regroupement des nœuds proposée dans cet article est statique et nécessite la connaissance de la matrice de délais. Nous envisageons l'utilisation d'un processus de regroupement distribué et auto-organisé comme proposé dans [MIN 06].

Remerciements

François Cantin et Bamba Gueye sont financés respectivement par le Fonds pour la Recherche dans l'Industrie et l'Agriculture (*F.R.I.A*), et par le projet européen *ANA*.

6. Bibliographie

[CHU 00] H. CHU Y., RAO S. G., ZHANG H., « A case for end system multicast », *Proc. the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Santa Clara, jun 2000.

[COS 04] COSTA M., CASTRO M., ROWSTRON R., KEY P., « PIC : practical Internet coordinates for distance estimation », *Proc. International Conference on Distributed Computing*

Systems, 2004, p. 178–187.

[DAB 04] DABEK F., COX R., KAASHOEK F., MORRIS R., « Vivaldi : A decentralized network coordinate system », *Proc. ACM SIGCOMM*, Portland, OR, USA, août 2004.

[GUM 02] GUMMADI K. P., SAROIU S., GRIBBLE S. D., « King : Estimating latency between arbitrary Internet end hosts », *ACM Internet Measurement Workshop 2002*, Marseille, France, novembre 2002.

[KUB 00] KUBIATOWICZ J., ET AL., « OceanStore : An Architecture for Global-Scale Persistent Storage », *Proc. the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, nov 2000.

[LED 07] LEDLIE J., GARDNER P., SELTZER M. I., « Network Coordinates in the Wild », *Proc NSDI*, Cambridge, apr 2007.

[LEE 06] LEE S., ZHANG Z., SAHU S., SAHA D., « On suitability of Euclidean embedding of internet hosts », *SIGMETRICS*, vol. 34, n° 1, 2006, p. 157–168, ACM Press.

[MIN 06] MIN S., HOLLIDAY J., CHO D., « Optimal Super-peer Selection for Large-scale P2P System », *Proc. the International Conference on Hybrid Information Technology*, Washington, DC, USA, nov 2006, p. 588–593.

[NG 02] NG T. S. E., ZHANG H., « Predicting Internet Network Distance with Coordinates-Based Approaches », *Proc. IEEE INFOCOM*, New York, NY, USA, juin 2002.

[P2P] « A simulator for peer-to-peer protocols », `http://www.pdos.lcs.mit.edu/p2psim/index.html`.

[PIA 03] PIAS M., CROWCROFT J., WILBUR S., HARRIS T., BHATTI S., « Lighthouses for Scalable Distributed Location », *Proc. the Second International Workshop on Peer-to-Peer Systems*, Berkeley, CA, USA, février 2003.

[ROW 01] ROWSTRON A., DRUSCHE P., « Pastry : Scalable, distributed object location and routing for large-scale peer-to-peer systems », *Proc. IFIP/ACM International Conference on Distributed Systems Platforms*, Heidelberg, Germany, Nov 2001.

[SAV 99] SAVAGE S., COLLINS A., HOFFMAN E., SNELL J., ANDERSON T., « The End-to-end Effects of Internet Path Selection », *Proc. of ACM SIGCOMM'99*, Cambridge, MA, USA, septembre 1999.

[SHA 03] SHAVITT Y., TANKEL T., « Big-Bang Simulation for Embedding Network Distances in Euclidean Space », *Proc. IEEE INFOCOM*, San Francisco, CA, USA, mars 2003.

[SKY] SKYPE, `www.skype.com/`.

[WAN 07] WANG G., ZHANG B., NG T. S. E., « Towards network triangle inequality violation aware distributed systems », *Proc. the 7th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2007, p. 175–188.

[Y.B 03] Y. BARTAL N. LINIAL M. M., NAOR A., « On metric ramsey-type phenomena », *Proc. the Annual ACM Symposium on Theory of Computing (STOC)*, San Diego, CA, june 2003.

[ZHE 05] ZHENG H., LUA E. K., PIAS M., GRIFFIN T., « Internet Routing Policies and Round-Trip-Times », *Proc. PAM Workshop*, Boston, MA, USA, avril 2005.

Conception de réseau iBGP

Marc-Olivier Buob, Steve Uhlig, Mickaël Meulle

France Télécom R&D
38-40, rue du Général Leclerc
92794 Issy les Moulineaux Cedex 9

{marcolivier.buob,michael.meulle,}@orange-ftgroup.com
S.P.W.G.Uhlig@tudelft.nl

RÉSUMÉ. *L'Internet est constitué de plus de 25,000 AS (Autonomous System) échangeant des informations de routage grâce à BGP (Border Gateway Protocol). Dans un AS de taille importante, il n'est pas possible d'établir une session BGP entre chaque paire de routeur pour des raisons de scalabilité. On a alors recours à la réflexion de route. Cependant, cette technique induit une opacité en terme de diffusion des routes, et peut provoquer l'apparition de routages sous-optimaux (en terme de coût IGP), des déflexions de routes, voire des boucles de routage. Dans ce travail nous proposons une solution pour construire une topologie de réflexion de route permettant d'avoir un routage identique à celui d'un full mesh iBGP, y compris en cas de panne simple d'équipement, et en installant un minimum de sessions iBGP. Nous avons appliqué cette méthode sur le réseau d'un opérateur tier-1 et calculé une topologie iBGP robuste à tout cas de panne simple. La topologie obtenue reste de taille comparable à celle actuellement déployée.*

ABSTRACT. *The Internet is made of more than 25,000 Autonomous Systems (AS) exchanging routing information with the Border Gateway Protocol (BGP). In large ASs, establishing a full-mesh of BGP sessions between the routers does not scale. Route-reflection was introduced to provide scalability. The drawback of route-reflection is the opacity in the diffusion of the routes, and may lead to suboptimal routing, deflection, and routing loops. In this work, we propose a solution to build a route-reflection topology that guarantees a routing equivalent to an iBGP full-mesh, even under simple IGP failures, and with a minimal number of iBGP sessions. We applied this method to the network of a tier-1 provider and computed an iBGP topology robust to simple IGP failures. The resulting topology has a similar size to the currently deployed one.*

MOTS-CLÉS : *BGP, reflexion de route, conception de topologie iBGP, optimisation*

KEYWORDS: *BGP, route-reflection, iBGP topology design, optimization*

1. Introduction

L'Internet rassemble plus de $25,000$ domaines interconnectés appelés Autonomous System (AS). Le routage à l'intérieur d'un domaine est assuré par un protocole IGP (Interior Gateway Protocol) [HAL 00], comme IS-IS ou OSPF. Ce protocole de routage permet de trouver les plus court chemins (au sens IGP) entre chaque paire de routeurs de l'AS. Les destinations extérieures à l'AS sont apprises grâce aux informations de routage BGP[REK 95]. Les routeurs implémentant le protocole BGP sont en mesure de savoir vers quel routeur envoyer le trafic pour joindre une destination extérieure à leur AS. Le trafic est routé vers ce point de sortie en accord avec le routage IGP. Si au cours de ce chemin, le point de sortie est modifié, on parle de déflexion.

Les informations de routage BGP sont échangées entre les AS au travers des sessions eBGP (External BGP). Ces sessions sont installées sur les liens inter-domaines (i.e. les liens interconnectant deux routeurs de bordure (ASBR) appartenant à des AS différents). Les informations de routage circulent ensuite dans chaque AS grâce à des sessions iBGP (Internal BGP). Avec iBGP classique, un routeur BGP ne retransmet jamais un message iBGP vers un autre voisin iBGP. Cette règle permet de réduire le nombre de messages BGP au sein de l'AS. Ainsi, chaque routeur doit établir une session iBGP vers chaque autre routeur de l'AS afin de propager ses routes à l'ensemble des routeurs de l'AS. Une telle topologie, appelée *full mesh iBGP*, requiert $n(n-1)/2$ sessions iBGP où n est le nombre de routeurs BGP présents dans l'AS. Cette solution, communément utilisée dans les petits AS, ne passe pas à l'échelle. En effet, chaque routeur va devoir maintenir une table par session, et devenir sensible au moindre évènement dans le réseau. De plus, l'ajout d'un nouvel équipement dans une telle topologie devient rapidement lourd à configurer puisqu'il nécessite une intervention sur chaque routeur BGP. C'est pourquoi dans un AS de taille importante, les opérateurs réseaux ont recours aux confédérations BGP[1] [TRA 01], ou à la réflexion de route[2].

Dans cet article, nous nous focalisons sur la réflexion de route car c'est la technique la plus communément utilisée dans les grands AS. Un réseau utilisant la réflexion de route peut être victime de problèmes de routage dans deux cas.

1) Certains routeurs choisissent leur route en accord avec le MED. Le Multi Exit Discriminator est un attribut BGP permettant d'implémenter une forme de "cold potato routing". Ces problèmes de routage sont étudiés dans [GRI 02a] et peuvent être facilement évités en déployant sur chaque routeur BGP l'option *always-compare-med* ou *set-deterministic-med*.

2) Les routeurs choisissent leur route sur une étape de décision BGP implémentant le "hot potato routing" : *préférer une route apprise par eBGP sur une route apprise par iBGP*, et *préférer les routes BGP dont le point de sortie est le plus proche au sens IGP* [BUO 07].

---

1. Ceci consiste à subdiviser un AS en plusieurs sous-AS
2. Certains routeurs BGP, appelés route reflector (RR), peuvent repropager une partie des messages iBGP qu'ils reçoivent à des voisins iBGP.

Dans un AS de taille importante, les étapes de processus de décision implémentant le “hot potato routing” sont fréquemment déterminantes [TEI 04] (70% des routes dans le réseau tier-1 que nous avons étudié). Cependant, la réflexion de route ne garantit pas qu’un routeur apprenne systématiquement son point de sortie optimal pour une destination donnée. Idéalement, le routage devrait cependant converger vers le même état que celui obtenu dans un full mesh. Une telle topologie iBGP est dite *fm-optimale* [BUO 07]. Cette propriété permet en particulier de garantir un routage optimal, déterministe, et sans déflexion ou boucle de routage.

Dans cet article, nous cherchons à construire une topologie iBGP vérifiant les critères suivants :

– **Fm-optimalité :** on souhaite concilier les avantages offerts par le full mesh iBGP (du point du vue routage) et de la réflexion de route (en terme de configuration et de charge au niveau des routeurs). En outre ce critère garantit que chaque routeur choisit son meilleur point de sortie possible pour n’importe quelle destination. Comme nous allons le voir par la suite, valider ce critère dans une topologie de RR n’est pas évident, mais possible.

– **Validité :** prouver que chaque paquet parvient à atteindre son point de sortie est un problème NP-difficile. Cependant, le concept de *fm-optimalité* garantit qu’un réseau ne comporte pas de déflexion de routage et donc pas de boucle de routage.

– **Reliabilité :** on construit une topologie iBGP aussi proche que possible de la topologie IGP [XIA 03, GRI 02b]. On évitera autant que possible les sessions iBGP multi-saut, i.e. les sessions entre routeurs qui ne sont pas directement adjacents dans le réseau.

– **Robustesse :** la topologie construite doit rester valide en cas de panne IGP. Dans cette approche, nous allons construire une topologie qui reste *fm-optimale* pour tout cas de panne IGP simple (lien ou routeur).

– **Passage à l’échelle :** on construit une topologie iBGP avec aussi peu de sessions que possible.

Sauf erreur, l’approche que nous proposons est la seule qui garantit un bon comportement du réseau pour tout cas de panne simple. Qui plus est, elle garantit que les routages restent optimaux lors de ces pannes. L’approche que nous proposons permet de traiter efficacement des instances de grands reseaux, en particulier des réseaux d’opérateurs tier-1.

La section 2 introduit les notations utilisées dans le reste de l’article. La section 3 présente la méthode de résolution utilisée. La section 4 valide cette approche d’une part sur des instances aléatoires, et d’autre part sur deux réseaux réels. L’état de l’art est présenté dans la section 5.

2. Terminologie

On note $G_{igp} = (V_{igp}, E_{igp})$ le graphe caractérisant la structure physique du réseau. Chaque sommet de V_{igp} représente un routeur, et chaque arc pondéré $(u, v) \in$

E_{igp} représente un lien physique et la métrique associée. On note $|u, v|$ la longueur du plus court chemin de u à v.

On note $\mathcal{N}$ l'ensemble des routeurs de bordure de l'AS, et $\mathcal{R}$ l'ensemble des routeurs implémentant le protocole BGP dans l'AS. En outre $\mathcal{N} \subseteq \mathcal{R}$. Le graphe $G_{bgp} = (V_{bgp}, E_{bgp})$ permet de représenter la topologie de réflexion de route déployée dans le réseau ($V_{bgp} = \mathcal{R}$). E_{bgp} rassemble l'ensemble des sessions iBGP installées. Chaque arc iBGP est étiqueté en accord avec les notations utilisées dans [FEA 04, GRI 02b]. Quand un routeur établit une session iBGP vers un autre routeur, on construit un arc étiqueté up (d'un client vers un RR), $down$ (d'un RR vers un client), ou $over$ entre deux routeurs (routeur de même niveau hiérarchique) (voir figure 1 et section 2).

Soit $\mathbb{L} = \{up, over, down\}$ l'ensemble des étiquettes iBGP, et $label : E_{bgp} \longrightarrow \mathbb{L}$ la fonction qui retourne l'étiquette d'un lien donné. Soit $sym : \mathbb{L} \longrightarrow \mathbb{L}$ la fonction qui retourne le label symétrique d'un label donné : $sym(up) = down$, $sym(down) = up$, $sym(over) = over$.

On dit qu'un chemin orienté de G_{bgp} est *valide* si celui-ci est constitué d'une suite de 0 ou plusieurs arcs up, suivi de 0 ou 1 arc $over$, suivi de 0 ou plusieurs arcs $down$.

A chaque fois que l'on considère une paire $(n, r) \in \mathcal{N} \times \mathcal{R}$ on suppose que :

– n est le plus proche ASBR de r au sens IGP pour un jeu de routes particulier ;

– la meilleure route choisie par chaque routeur est choisie sur le critère "privilégier le plus proche point de sortie appris" ;

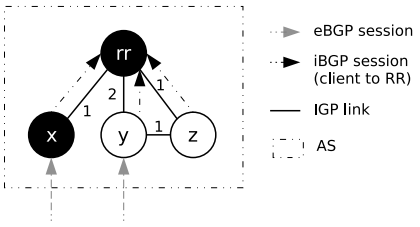
– il existe un préfixe destination p annoncé par plusieurs routes concurrentes reçues via eBGP par des ASBR différents.

On cherche à garantir que r est toujours capable d'apprendre la route annoncée par son meilleur point de sortie n. Les points de sortie concurrents sous-optimaux sont donc décrits par l'ensemble : $\mathcal{N}(n, r) = \{n' \in \mathcal{N}, |r, n'| > |r, n|\}$.

S'il existe au moins un chemin iBGP valide de n à r tel que chaque routeur w de ce chemin choisit la route annoncée par n, alors r choisit la route annoncée par n. On appelle *routeur blanc* un routeur w vérifiant cette propriété. Ainsi, l'ensemble des routeurs blancs relatifs à une paire (n, r) se définit par : $\mathcal{W}(n, r) = \{w \in \mathcal{R} | \forall n' \in \mathcal{N}(n, r), |r', n| < |r', n'|\}$

On appelle *chemin blanc* tout chemin iBGP uniquement constitué de routeurs blancs. Si pour toute paire $(n, r) \in \mathcal{N} \times \mathcal{R}$, il existe au moins un chemin valide blanc, alors G_{bgp} est *fm-optimal*. On remarque en particulier que le concept de *fm-optimalité* est indépendant de la notion de préfixe. Ainsi, ce critère garantit un bon comportement du réseau pour *tout* jeu de routes BGP concurrentes.

La figure 1 illustre les concepts que nous venons d'introduire. Dans cet exemple, (y, rr, z) est un chemin iBGP valide de y à z. Cependant $rr \notin \mathcal{W}(y, z)$, donc (y, rr, z) n'est pas un chemin blanc. En effet, rr choisit la route annoncée par x (si x en reçoit une). Ainsi rr est susceptible de ne pas propager la route annoncée par y vers z.



$(n, r) = (y, z)$
$\mathcal{N}(y, z) = \{x\}$
$\mathcal{W}(y, z) = \{y, z\}$

Figure 1 – Un exemple de routage sous-optimal : le trafic émis par z suit le chemin IGP (z, rr, x) au lieu de (z, y).

3. Comment construire une topologie iBGP fm-optimale ?

Nous allons détailler dans cette partie comment résoudre le problème de conception iBGP que nous avons défini. Les paramètres requis pour ce problème sont l'ensemble des ASBR ($\mathcal{N}$), l'ensemble des routeurs BGP ($\mathcal{R}$), et la topologie IGP (G_{igp}). Notre approche part du principe que $V_{igp} = V_{bgp}$ ce qui est généralement le cas en pratique, et calcule un ensemble idéal de sessions iBGP (E_{bgp}). On modélise ce problème sous forme d'un programme linéaire en nombre entier (PLNE). Cependant, les contraintes ne pouvent pas être énumérées de manière exhaustive pour un réseau de taille importante. On génère donc dynamiquement les contraintes utiles au problème (décomposition de Benders).

1) Dans un premier temps nous n'allons pas tenir compte des pannes (section 3.1). On s'intéresse donc uniquement au comportement du réseau dans son *régime nominal*. Pour chaque paire $(n, r) \in \mathcal{N} \times \mathcal{R}$, on construit un problème satellite, satisfait si et seulement si un chemin valide blanc existe de n à r. Les problèmes satellites permettent d'alimenter le jeu de contraintes du PLNE (problème maître).

2) Dans un second temps (section 3.2), on explique comment introduire les contraintes de robustesses aux pannes. On construit pour cela autant de satellites (n, r, f) (où f désigne une panne) que nécessaire. Il est nécessaire d'agréger les satellites ainsi engendrés, sans quoi le problème est trop grand pour être résolu.

Nous n'allons pas considérer les sessions $over$ dans ce problème, bien que notre modèle le permette. Ceci permet en outre de limiter le phénomène de dégénérescence du problème. En effet, chaque session $over$ peut être transformée en session up ou $down$ sans remettre en cause la validité de la topologie calculée.

3.1. *Cas nominal*

3.1.1. *Problème maître*

Pour chaque session iBGP candidate (u, v), $(u, v) \in \mathcal{R}, u \neq v$, on définit deux variables booléennes : $up(u, v)$ (égale à 1 si $label(u, v) = up$, 0 sinon) ; $down(u, v)$ (égale à 1 if $label(u, v) = down$, 0 sinon).

On définit une fonction objectif F permettant de construire une topologie iBGP aussi proche que possible de la topologie IGP, tout en minimisant le nombre de session iBGP installées :

$$F = min(\sum_{(u,v)\in\mathcal{R}} (R(u,v).(up(u,v) + down(u,v))))$$

où $R(u,v)$ est égal au nombre de sauts IGP nécessaires pour établir une session iBGP de u à v.

Le problème maître comporte deux types de contraintes :

– **Les contraintes de domaine :** chaque paire de routeur $(u,v) \in \mathcal{R}\times\mathcal{R}$ est reliée par au plus 1 session iBGP. De plus $label(u,v) = sym(label(v,u))$. Pour chaque session iBGP candidate (u,v) on introduit les contraintes suivantes :

- $\forall u,v \in \mathcal{R},\ up(u,v) + down(u,v) \leq 1$,
- $\forall u,v \in \mathcal{R},\ up(u,v) = down(v,u)$.

– **Les contraintes Max-flow Min-cut :** au début, cet ensemble de contrainte est vide. Il sera alimenté à chaque itération par les problèmes satellites (voir section 3.1.2).

A chaque itération it, le problème maître est résolu et propose aux problèmes satellites la solution qu'il vient de calculer. Chaque satellite violé remonte une contrainte Max-flow Min-cut qui sera insérée dans le problème maître. Au fil des itérations, le jeu de contraintes s'enrichit jusqu'à ce que l'ensemble des satellites soient satisfaits. Dès lors, il existe pour chaque paire (n,r) au moins un chemin blanc valide. Cette ultime résolution du problème maître permet de calculer une topologie *fm-optimale* tout en minimisant la fonction objectif F.

3.1.2. *Problèmes satellites*

Afin de ne construire que des chemins iBGP valides, on réutilise la transformation de graphe proposée dans [BUO 07]. Chaque sommet de V_{bgp} est transformé en un *meta noeud* composé de deux sommets (appelés *noeud source* et *noeud cible*) et un arc (appelé *arc interne*), comme montré sur la figure 2. La manière dont deux *meta noeuds* sont connectés découle directement du type de session iBGP établie entre les deux routeurs correspondants. Dans le graphe étendu, on ne peut construire que des chemins valides. On appelle *meta arc* tout arc connectant deux sommets appartenant à deux *meta noeuds* différents. Chaque *meta arc* correspond à un type de session iBGP établie entre deux routeurs. On note $[u,v,rel]$ le *meta arc* allant du *meta noeud* u au *meta noeud* v et portant la relation iBGP rel. Chaque chemin valide de G_{bgp} allant de $s \in V_{bgp}$ à $t \in V_{bgp}$ correspond ainsi à un chemin du graphe étendu allant de s_{src} à t_{dst}, où s_{src} est le *noeud source* de s et t_{dst} le *noeud cible* de t.

Un problème satellite permet de vérifier si un chemin valide blanc existe pour une paire (n,r) donnée. Un tel problème peut se modéliser à l'aide d'un problème de flot dans un graphe noté $G_w(n,r)$. Chaque sommet de ce graphe appartient à $\mathcal{W}(n,r)$ (voir section 2). Afin de réduire le nombre de sessions iBGP candidates, on considère uniquement les sessions iBGP (u,v) vérifiant la propriété $|n,u| \leq |n,v|$ et

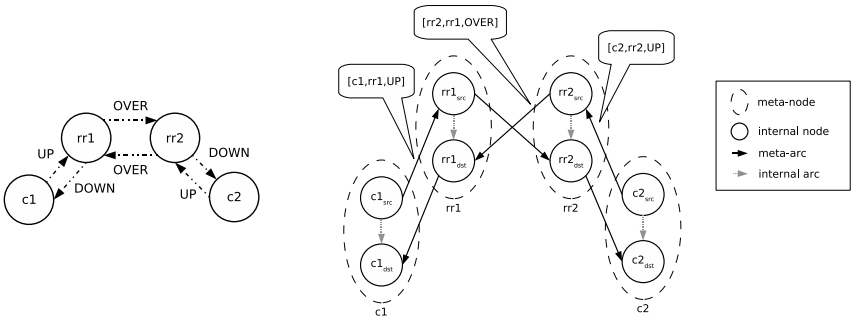


Figure 2 – Un exemple de graphe iBGP et son graphe étendu correspondant. Le chemin iBGP valide (c_1, rr_1, rr_2, c_2) dans G_{bgp} correspond au chemin $(c1_{src}, rr1_{src}, rr2_{src}, rr2_{dst}, c2_{dst})$ dans G^{ext}_{bgp}.

$|v, r| \leq |u, r|$. Ainsi, un message BGP se rapproche toujours de sa destination r et s'éloigne de sa source n au cours de la propagation iBGP. Ceci permet d'acheminer plus rapidement les messages de n vers r. Le graphe $G_w(n, r) = (\mathcal{W}(n, r), E_w(n, r))$ regroupe donc l'ensemble des chemins iBGP blancs aptes à satisfaire efficacement une paire (n, r) donnée.

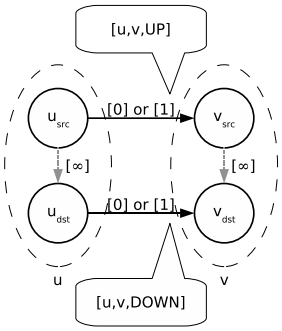


Figure 3 – Deux sessions iBGP candidates : $label(u, v) = up$ ou $label(u, v) = down$.

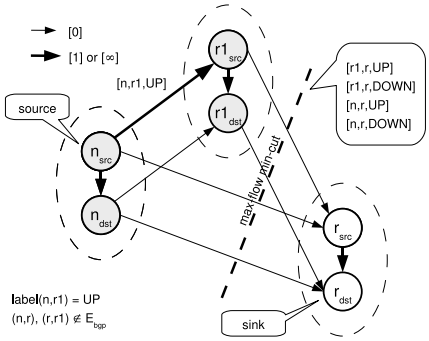


Figure 4 – La coupe min-cut max-flow engendrée : $up(r_1, r) + down(r_1, r) + up(n, r) + down(n, r) \geq 1$.

On construit ensuite le graphe étendu $G^{ext}_w(n, r)$ correspondant. On note n_{src} le *noeud source* du *meta noeud* n et r_{dst} le *noeud cible* du *meta noeud* r. On installe l'ensemble des *meta arcs* correspondant à l'ensemble des sessions iBGP candidates. La capacité installée sur chaque meta arc (i, j) dépend des sessions installées :

– Si i et j appartiennent au même *meta noeud*, on installe une capacité infinie.

– Dans le cas contraire, on installe une capacité nulle ou égale à 1. Soient $rel \in$

$\{up, down\}$ le type de session iBGP correspondant à l'arc (i, j) et r_i (resp. r_j) le *meta-noeud* correspondant à i (resp. j). Si une session iBGP de type rel est installée de r_i à r_j, on installe sur (i, j) une capacité égale à 1 (0 autrement). En outre, au plus un *meta arc* du *meta noeud* r_i au *meta noeud* r_j est de capacité égale à 1.

Si le flot maximal de n_{src} (la source) à r_{dst} (la destination) est supérieur ou égal à 1, alors la paire (n, r) est satisfaite. Autrement aucune unité de flot ne parvient à atteindre la destination. Dans ce cas, on cherche la coupe minimale de flot maximale. On note $C(n, r, it)$ l'ensemble des *meta arcs* coupés durant l'itération. On insère ensuite dans le problème maître la contrainte linéaire max-flow min-cut :

$$\sum_{[r_i, r_j, rel] \in C(n,r,it)} (rel(r_i, r_j)) \geq 1.$$

La figure 4 permet de visualiser une telle coupe. Dans cet exemple $\mathcal{W}(n, r) = \{n, r_1, r\}$. Les itérations précédentes ont conduit à installer une session up de n vers r_1 et pas de session entre r_1 et r, et entre n et r. Le satellite (n, r) est ici violé. Il engendre la contrainte $up(r_1, r) + down(r_1, r) + up(n, r) + down(n, r) \geq 1$, insérée à l'itération suivante dans le problème maître.

3.2. *Cas des pannes IGP*

Cette section explique comment tenir compte des pannes IGP (de lien ou de routeur). Rappelons qu'une session BGP s'établit le long du plus court chemin IGP du routeur source au routeur destination de la session. Lorsqu'une panne IGP survient, chaque routeur met à jour ses plus courts chemins IGP et, dans le cas où la panne a impacté certaines sessions iBGP ces dernières sont remontées automatiquement. Si la connectivité IGP est perdue entre les deux routeurs BGP, la session BGP tombe. On note une panne IGP f une panne impactant un ou plusieurs équipements (lien ou routeur), et ϕ la panne vide.

Pour chaque paire f on recalcule le coût IGP entre chaque paire de routeur appartenant à une même composante connexe IGP. Il suffit ensuite d'appliquer le raisonnement du cas nominal au sein de chacune de ces composantes connexes. Soit une paire (n, r) telle que n et r appartiennent à une même composante connexe IGP C. On considère dans $G_w(n, r, f)$ uniquement les sommets blancs appartenant à C. En effet, une session iBGP ne peut s'établir que si les deux routeurs appartiennent à la même composante connexe IGP. Pour chaque couple (n, r) on se limite donc aux pannes f telles que n et r appartiennent toujours à la même composante connexe, et n'impactant ni n, ni r. Ce formalisme est suffisamment générique pour prendre en compte n'importe quel jeu de pannes simples ou multiples passé en paramètre du problème.

Si l'on construit tous les satellites (n, r, f), on constate que certains sont redondants. Par exemple, si f n'affecte en rien la paire (n, r), alors il est inutile d'introduire le problème satellite (n, r, f) dans le problème. En effet, celui-ci introduira les mêmes contraintes que (n, r, ϕ). De manière plus générale, considérons deux pannes f, f' et

une paire (n, r). Soient $G_w(n, r, f)$ et $G_w(n, r, f')$ les deux graphes correspondants. Si $G_w(n, r, f) \subseteq G_w(n, r, f')$, alors les contraintes introduites par $G_w(n, r, f)$ seront toujours moins restrictives que celles introduites par $G_w(n, r, f')$. On peut donc supprimer sans risque le problème satellite (n, r, f').

4. Résultats

Nous avons d'abord appliqué notre approche sur de petites topologies (voir section 4.1) dont une celle du réseau GEANT. Nous avons ensuite résolu le problème sur le réseau d'un grand opérateur tier-1 (section 4.2). Nous avons calculé pour chaque instance calculé deux topologies iBGP : la première ne tient pas compte des cas de panne, la seconde est robuste à tout cas de panne simple IGP.

4.1. *Petites topologies*

Cette partie présente les résultats obtenus dans le cas nominal et le cas robuste aux pannes simples sur 5 topologies de petite taille :

– La topologie du réseau GEANT de 2004[3].

– 4 topologies générées par iGen [4]. iGen permet de générer un ensemble de points aléatoire sur un ou plusieurs continents, et les connecte en utilisant différentes heuristiques de conception de réseau [CAH 98]. Nous avons généré 4 réseaux de 25 noeuds. NA correspond à une topologie IGP de réseau pour le continent Nord américain, W à une topologie mondiale. Nous avons utilisé deux heuristiques de maillage : la triangulation de Delaunay (D) et une juxtaposition de deux arbres couvrants disjoints (2T).

Afin de se placer dans le cas le plus difficile, on suppose que chaque routeur est un routeur de bordure ($\mathcal{N} = \mathcal{R} = V_{igp}$). C'est la forme la plus contrainte du problème. En effet, les topologies iBGP calculée restent *fm-optimales* même si seuls certains routeurs sont effectivement des routeurs de bordure. Plus précisément, une topologie iBGP calculée pour un plus petit jeu d'ASBR comporterait moins de session iBGP. Le tableau 1 présente les résultats obtenus avec et sans prise en compte des pannes (voir les deux lignes du bas) pour les 5 topologies que nous venons de présenter.

GEANT utilise un full mesh iBGP et requiert la configuration de 462 sessions iBGP orientées entre ses 22 routeurs. La décomposition de Benders montre que 74 sessions pourrait suffire pour obtenir le même routage dans le cas nominal, et 170 pour garantir la *fm-optimalité* du réseau en cas de panne simple. Le nombre de session iBGP double lorsque la topologie doit rester robuste aux pannes simples IGP (voir dernière ligne du tableau 1). Néanmoins les topologies ainsi construites comportent 3 fois moins de sessions que le full mesh iBGP correspondant.

3. http://www.geant.net
4. http://www.info.ucl.ac.be/~bqu/igen/

		GEANT	NA-D	NA-2T	W-D	W-2T
Graphe d'entrée	$\|V_{bgp}\| = \|V_{igp}\|$	22	25	25	25	25
	$\|E_{igp}\|$	72	128	96	130	96
	$\|E_{bgp}\|$ in f.m.	462	600	600	600	600
Sans panne	$\|E_{bgp}\|$	74	80	72	100	64
Avec panne	$\|E_{bgp}\|$	172	168	146	194	126

Tableau 1 – Les solutions trouvées pour les petites topologies.

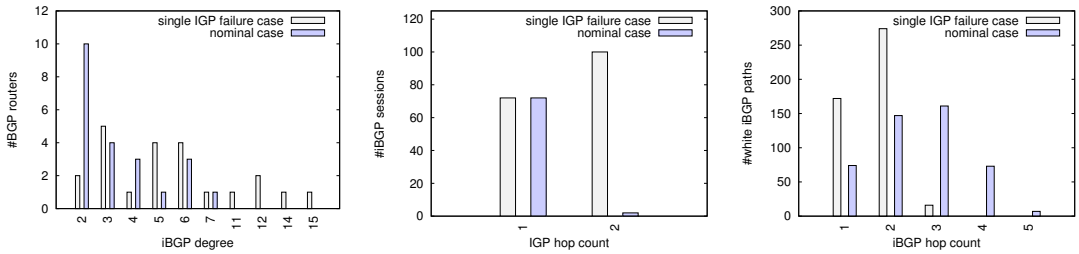


Figure 5 – Les propriétés des topologies iBGP calculées pour le réseau GEANT (cas nominal et cas robuste aux pannes simples).

Afin de caractériser la qualité des topologies calculées, nous utilisons trois indicateurs. Nous n'avons reporté que les résultats pour le réseau GEANT, car les topologies IGEN conduisent à des observations similaires (voir figure 5)

1) Distribution du degré de chaque sommet iBGP : lorsqu'un routeur établit un nombre important de session iBGP, une quantité importante de mémoire est consommée (une table par session). Le graphe de gauche indique qu'un routeur a au plus 7 voisins iBGP dans le cas nominal, contre 15 dans le cas robuste aux pannes simples.

2) Distribution de la longueur de chaque chemin blanc : nous avons calculé pour chaque paire source destination (n, r) le nombre de sauts iBGP requis. De cette valeur découle la rapidité à laquelle les messages iBGP seront traités et diffusés. Ainsi plus un chemin est long, moins l'information se diffuse rapidement. Le graphe du milieu reporté montre que dans la plupart des chemins iBGP comportent moins de deux bonds iBGP.

3) Concordance avec la topologie IGP : pour chaque session iBGP, on calcule le nombre de routeurs traversés. Idéalement la topologie iBGP devrait être aussi proche que possible de la topologie IGP [XIA 03, GRI 02b] (soit un seul bond). Le graphe de droite montre que c'est le cas pour la très grande majorité des sessions iBGP, en particulier pour le cas nominal.

4.2. *Application sur le réseau de coeur d'un opérateur tier-1*

Nous avons appliqué notre méthode sur le réseau d'un opérateur tier-1 composé de plusieurs centaines de routeurs BGP. Les topologies calculées sont assez éloignées

de la topologie utilisée. Ceci sous-entend donc une migration complexe à mettre en oeuvre, mais ce n'est pas très surprenant. En effet, la règle de conception couramment utilisée (3 étages hiérarchiques de RR, avec en haut de la topologie les routeurs inter-continentaux, ensuite les routeurs continentaux, et en bas les routeurs nationaux) est simple à appréhender, mais ne conduit pas à une topologie iBGP robuste et efficace.

On remarque que la topologie correspondant au cas nominal comporte 45% moins de session que la solution déployée dans le réseau et elle est en plus *fm-optimale*. En outre, chaque routeur établit moins de sessions iBGP. Les chemins blancs sont un peu plus longs que dans réseau actuel. La topologie robuste comporte 25% sessions supplémentaires à celle déployée dans le réseau, mais elle est *fm-optimale* y compris en cas de panne simple. La distribution des degrés des routeurs iBGP et la distribution des chemins blancs est proche de la solution déployée dans le réseau.

5. Etat de l'art

[DUB 99] est le premier article à mettre en évidence qu'un réseau peut être victime de boucles de routage si la topologie iBGP est mal définie. [GRI 02b] propose des conditions permettant d'éviter l'apparition de boucles de routage : 1) les RR devraient préférer les routes apprises par leur clients sur les autres, 2) chaque plus court chemin IGP devrait être valide au sens iBGP. Ces deux conditions sont cependant très restrictives. [VUT 06] propose une approche permettant de s'affranchir de la première condition.

[XIA 03] propose une formulation du problème de conception iBGP. Les auteurs introduisent deux critères (Expected Lifetime et Expected Session Loss) apportant une certaine forme de robustesse. Ils construisent ensuite une topologie iBGP sur deux niveaux hiérarchiques tout en optimisant ces deux critères. [VUT 06] propose une méthode de séparation de graphe permettant de construire itérativement des étages de RR. La topologie ainsi construite ne comporte pas de boucle de routage en régime nominal. [RAW 06] présente en détail les problèmes de routages communément rencontrés dans un réseau utilisant la réflexion de route. Cet article propose quelques conditions permettant d'éviter l'apparition de déflexion de route et d'oscillations de routage dûes au MED. La méthode proposée permet de construire une topologie de RR à deux niveaux et minimise la distance IGP entre deux voisins iBGP.

Aucune de ces approches ne permet de garantir que la topologie iBGP reste valide en cas de panne simple IGP. En outre une telle panne pourrait provoquer l'apparition de boucle de routage. De plus, seul [VUT 06] permet de construire des topologies composées de plus de deux étages de RR.

6. Conclusion

Dans cet article nous avons proposé une méthode permettant de construire une topologie iBGP de réflexion de route. D'une part, notre topologie passe à l'échelle en tirant parti de la réflexion de route. D'autre part, le réseau se comporte comme si un full mesh avait été déployé et apporte les mêmes garanties de routage, y compris en

cas de panne simple d'équipement. Notre approche est la première à apporter de telles garanties. Globalement, assurer la robustesse à chaque cas de panne simple requiert l'installation de deux fois plus de sessions iBGP.

Nous avons vu qu'iBGP diffuse les informations de routage dans un AS. Cette diffusion dépend d'une part du graphe de diffusion (donc du placement des sessions iBGP) et du mécanisme de propagation utilisé. Nous avons ici cherché à résoudre les problèmes de routage en optimisant ce graphe de diffusion. Cependant, ce graphe est fortement dépendant de la topologie IGP. Idéalement, le protocole iBGP ne devrait pas induire de telles difficultés de conception. Nous cherchons actuellement à modifier le protocole iBGP lui même afin de pouvoir configurer un réseau iBGP à l'aide de règles de conception simples. Nous tenons à remercier Olivier Klopfenstein et Jean - Luc Lutton pour leur aide précieuse.

7. Bibliographie

[BUO 07] BUOB M., MEULLE M., UHLIG S., « Checking for optimal egress points in iBGP routing », Proc. of the 6th IEEE International Workshop on the Design of Reliable Communication Networks (DRCN 2007), October 2007.

[CAH 98] CAHN R., *Wide area network design : concepts and tools for optimization*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[DUB 99] DUBE R., « A comparison of scaling techniques for BGP », *SIGCOMM Comput. Commun. Rev.*, vol. 29, n° 3, 1999, p. 44–46.

[FEA 04] FEAMSTER N., WINICK J., REXFORD J., « A Model of BGP Routing for Network Engineering », *ACM Sigmetrics - Performance 2004*, New York, NY, June 2004.

[GRI 02a] GRIFFIN T., WILFONG G. T., « Analysis of the MED Oscillation Problem in BGP », *ICNP '02 : Proceedings of the 10th IEEE International Conference on Network Protocols*, Washington, DC, USA, 2002.

[GRI 02b] GRIFFIN T. G., WILFONG G., « On the correctness of iBGP configuration », *Proc. of ACM SIGCOMM*, August 2002.

[HAL 00] HALABI B., PHERSON D. M., *Internet Routing Architectures (2nd Edition)*, Cisco Press, January 2000.

[RAW 06] RAWAT A., SHAYMAN M. A., « Preventing persistent oscillations and loops in IBGP configuration with route reflection », *Computer Networks*, , 2006, p. 3642-3665.

[REK 95] REKHTER Y., LI T., « A Border Gateway Protocol 4 (BGP-4) », RFC 1771, 1995.

[TEI 04] TEIXEIRA R., GRIFFIN T., VOELKER G., SHAIKH A., « Network Sensitivity to Hot Potato Disruptions », *Proc. of ACM SIGCOMM*, August 2004.

[TRA 01] TRAINA P., MCPHERSON D., SCUDDER J., « Autonomous System Confederations for BGP », RFC 3065, February 2001.

[VUT 06] VUTUKURU M., VALIANT P., KOPPARTY S., BALAKRISHNAN H., « How to Construct a Correct and Scalable iBGP Configuration », *IEEE INFOCOM*, Barcelona, Spain, April 2006.

[XIA 03] XIAO L., WANG J., NAHRSTEDT K., « Optimizing iBGP route reflection network », *IEEE INFOCOM*, 2003.

Origine des étoiles dans traceroute

Farah Layouni, Brice Augustin, Timur Friedman, Renata Teixeira
Université Pierre et Marie Curie, Laboratoire LIP6–CNRS

Laboratoire LIP6
104 avenue du président Kennedy 75016 Paris
{prenom.nom}@lip6.fr

RÉSUMÉ. *Traceroute permet de découvrir le chemin entre deux machines sur internet, en envoyant des sondes avec une durée de vie limitée pour forcer les routeurs à révéler leur présence. Lorsqu'il ne parvient pas à déterminer l'adresse IP d'un routeur, traceroute affiche à la place une étoile (*). Les nombreuses étoiles présentes dans les traces introduisent des imprécisions indésirables dans les applications de cartographie de l'internet, ou encore lors d'un diagnostic de panne. Cet article se penche sur les raisons de leur apparition, que nous classons en deux catégories. (1) Le routeur émet une réponse altérée qui empêche traceroute de l'associer à une sonde émise. (2) Le routeur n'émet tout simplement pas de réponse. Nous proposons des mesures qui mettent en évidence l'existence de chacune de ces causes. De plus, nous montrons comment réduire l'effet de ces étoiles grâce à un paramétrage approprié de l'outil de traçage.*

ABSTRACT. *Traceroute is a tool to measure the route between two machines in the internet. It sends TTL-limited probes to force routers to send error messages and reveal their presence. However, it frequently fails at discovering the routers' address at some portion of the path, and prints a star (*) instead. Those stars can appear for many reasons, and we classify them into two categories. (1) The router sends a corrupted response, which prevents traceroute to assign it to a probe it sent. (2) The router simply does not send any response. We propose measurements to characterize all stars in traces collected from our university. We also show that simple solutions can help preventing some of these causes, like an appropriate setup of the tool, or a simple modification of its mechanism.*

MOTS-CLÉS : *traceroute, mesures, internet, étoiles*

KEYWORDS: *traceroute, measurements, internet, stars*

1. Introduction

Traceroute est un outil pour mesurer le chemin entre deux hôtes d'un réseau IP. Il est largement utilisé dans le diagnostic de pannes, l'inférence de propriétés et la construction de cartes de l'internet. Il découvre les routeurs sur le chemin en émettant une série de paquets avec une durée de vie croissante. Cette durée de vie ("Time to Live", ou TTL) représente le nombre maximum de routeurs que le paquet peut traverser avant d'être détruit. Lorsqu'un routeur reçoit un paquet IP, il commence par décrémenter le TTL. S'il est inférieur à 1, alors le paquet a expiré et doit être détruit. Le routeur notifie l'émetteur du paquet rejeté via un message ICMP "TTL expiré" avec comme adresse source, l'adresse d'une de ses interfaces. Traceroute envoie des paquets (appelés sondes) à faible TTL pour forcer les routeurs à émettre ce type de message et ainsi révéler leur présence.

Il arrive fréquemment que traceroute ne reçoive pas de réponse de certains routeurs, et par conséquent ne parvienne pas à déterminer leur adresse. A la place il affiche une étoile (*), dont la présence pose de nombreux problèmes. Par exemple, elles empêchent un opérateur réseau de localiser précisément une panne ; elles introduisent des erreurs lors de la construction de cartes du réseau, certaines zones restant indéterminées ; elles allongent considérablement le temps de traçage, traceroute devant attendre l'expiration d'un temporisateur avant de continuer son exploration.

Cet article est la première étude systématique des causes de ces étoiles. Nous proposons des expériences pour mettre en évidence leur existence. Nous montrons que ces causes sont nombreuses et peuvent être classées en deux catégories. Premièrement, la réponse peut parvenir à l'émetteur, mais celui-ci la rejette parce qu'il est incapable de l'associer à une sonde. Cet échec d'association vient du fait que traceroute–et les outils dérivés–utilisent diverses méthodes ad hoc, non standardisées pour identifier les sondes qu'ils envoient. Nous discutons des différents types d'altération des réponses à l'origine de ce problème dans la section 2. Deuxièmement, certains équipements n'émettent tout simplement pas de réponse. Dans la section 3 nous montrons qu'il existe une grande variété de raisons pour lesquelles un routeur ou un hôte déciderait de rester muet. Enfin, nous proposons des bilans qui montrent comment utiliser ces résultats pour mieux paramétrer nos outils de traçage.

Nous proposons des expériences pour identifier chaque phénomène mis en cause. Ces expériences impliquent des protocoles de mesure différents, que nous décrirons au moment opportun. Néanmoins, nous avons utilisé une liste de 5000 adresses générées aléatoirement et répondant au ping au moment de la construction de la liste. Nos mesures consistent généralement à lancer un traceroute vers chaque destination, à partir d'une unique source située au Laboratoire d'Informatique de Paris 6, et à faire varier certains paramètres de mesure. Nos traces couvrent 1077 systèmes autonomes, ce qui représente une très faible portion du nombre total actuel. Néanmoins nous traversons la plupart des réseaux qui composent le coeur de l'internet, avec tous les tier-1 et 64 des 100 plus gros réseaux mondiaux Notre but n'est pas d'obtenir des statistiques représentatives de l'internet actuel, sur la prépondérance de chaque cause d'étoile. Nous

cherchons plutôt à montrer leur existence et proposer des expériences qui les mettent en évidence.

2. Échecs d'associations des réponses

Traceroute doit associer chaque réponse à une sonde émise. Pour cela il place un identifiant unique dans une partie de la sonde qui est recopiée dans le message d'erreur ICMP. Traceroute opère ensuite un tri sur tous les paquets ICMP reçus, pour déterminer les réponses qui lui sont destinées. Traceroute classique utilise les numéros de ports, technique rendue obsolète par le déploiement de répartiteurs de charge par flot dans l'internet [AUG 06]. Les contraintes sont donc nombreuses pour choisir le champ approprié où placer cet identifiant, et chaque outil utilise sa propre technique. La phase d'association de réponses est une phase critique lors du traçage, car ce champ peut subir des altération lors de son traitement par des noeuds du réseau. Si l'outil n'est pas capable d'identifier une réponse lui appartenant, il peut rejeter une réponse valide (ce qui cause l'apparition d'une étoile), et manque la possibilité de découvrir un nouveau noeud dans le réseau. Il peut aussi accepter une réponse invalide, d'où l'inférence de faux liens et le calcul de délais erronés.

L'expérience menée dans cette section permet de révéler ces altérations. Nous traçons nos 5000 destinations en ICMP, UDP et TCP, et capturons tous les paquets avec `tcpdump`. Nous analysons ensuite cette trace et associons chaque réponse avec une sonde. Comme certains champs peuvent être altérés, nous associons la réponse avec la sonde qui a le plus grand nombre de champs identiques : numéros de ports, identifiant IP, checksum UDP, séquence ICMP, séquence TCP. Cette association lâche nous permet de détecter tout type d'altération dans les réponses, que nous reportons ici.

2.1. *Inversion ou réécriture de l'identifiant IP*

Pour communiquer à travers Internet, deux machines doivent utiliser un format d'encodage précis, le "Network Byte Order", qui définit l'ordre dans lequel sont représentés les octets dans le réseau. Cet ordre peut être différent de celui utilisé en interne par une machine, c'est pourquoi chaque machine doit éventuellement effectuer une conversion avant d'émettre ou recevoir des données. Certaines implémentations inversent les octets de l'identifiant IP à la réception d'un paquet, puis l'incluent ainsi inversé dans un message d'erreur ICMP envoyé à la source [MAL 07]. Par conséquent, la source reçoit une réponse comportant un identifiant erroné. Tcptraceroute utilisant ce champ, il est incapable d'associer des réponses ainsi altérées. Dans nos traces nous avons détecté 251 réponses avec un identifiant inversé. Le traçage du chemin vers 31 destinations était affecté par ce problème. Plus rare encore (un unique cas dans nos traces), une seconde implémentation effectue une copie incorrecte du paquet IP en erreur, d'où un identifiant IP contenant des données aléatoires ou remises à zéro.

2.2. *Réécriture du checksum UDP*

La somme de contrôle UDP est un champ de 16 bits utilisé pour vérifier l'intégrité de l'entête et des données UDP. Certaines implémentations modifient ce champ avant de l'inclure dans un message d'erreur ICMP. Dans nos traces, 716 réponses comportent un checksum incorrect, soit 239 chemins affectés par ce problème. Ce problème étant largement répandu, l'utilisation de ce champ est à éviter (c'est le cas de Paris traceroute, qui l'utilise pour identifier les réponses aux sondes UDP).

2.3. *Réécriture de l'adresse destination*

Le passage de passerelles effectuant une translation d'adresse altère les paquets, et a un impact sur l'association des réponses. En effet, certaines passerelles modifient l'adresse destination des paquets entrants, avant de les transférer à une machine du réseau interne, mais oublient de faire la modification inverse dans les paquets d'erreur ICMP. En conséquence, l'outil de traçage reçoit une réponse altérée ; l'adresse destination du paquet original inclus dans la réponse (l'adresse d'une machine privée) est différente de l'adresse destination utilisée lors de l'émission de la sonde (adresse publique de la passerelle NAT). A noter que les adresses trouvées dans ces réponses altérées appartiennent principalement à des blocs d'adresses privées (192.168.0.0/16 et 10.0.0.0/8), ce qui confirme la présence d'une passerelle NAT masquant la structure du réseau interne.

Nos traces ont révélé 41 réponses avec une adresse originale altérée, soit 12 chemins affectés par ce problème. Paris traceroute vérifie la consistance de cette adresse avant d'accepter une réponse et rejette toute réponse avec une adresse différente de l'adresse originale. En conséquence, il rejette certaines réponses parfaitement valides.

Bilan : association des réponses. Il semble difficile d'élaborer une technique d'association résistante aux altérations et respectant toutes les contraintes. Tcptraceroute souffre du problème de l'altération de l'identifiant IP, ce qui n'est pas le cas de Paris traceroute pour TCP (utilisation du numéro séquence TCP) et ICMP (séquence et identifiant ICMP). Une solution pour utiliser sans risque l'identifiant IP est de n'utiliser que la moitié des identifiants disponibles, de manière à empêcher l'utilisation simultanée de deux identifiants qui pourraient être confondus s'ils étaient inversés (par exemple, 0x00ff et 0xff00).

2.4. *RTT très élevés*

Tous les outils de traçage attendent une réponse pendant quelques secondes puis abandonnent. Une attente plus longue semble inutile, dans la mesure où un temps d'aller-retour de plus de 5 secondes semble très improbable. La traversée de liens transocéaniques, ou encore la faible priorité accordée au trafic ICMP dans les routeurs modernes ne peut expliquer des délais de plusieurs secondes [CHO 04]. L'analyse

des traces de paquets nous a permis de calculer les temps d'aller-retour de toutes les réponses, y compris celles éventuellement arrivées en retard et donc manquées par l'outil de traçage. Sur 263566 réponses, 99.98% sont arrivées moins de 2 secondes après l'envoi de la sonde. 59 réponses sont arrivées entre 2 et 6 secondes après l'envoi. Par défaut, le délai d'abandon de Paris traceroute est de 2 secondes, ce qui lui ferait manquer toutes ces réponses, donnant au total 22 chemins affectés par ce problème. Notons que la majorité des réponses à délai élevé provient des extrémités du réseau. 70% sont des messages "Host unreachable" ou "Port Unreachable". Les premiers présentent souvent des délais d'environ 3 secondes, ce qui peut correspondre au délai d'attente, par une passerelle d'accès à Internet, d'une réponse ARP sur un LAN.

Plus inattendu, une petit nombre de réponses présentent des RTT anormalement élevés, jusqu'à 37 secondes. Plusieurs hypothèses peuvent expliquer ces délais. Premièrement, il est possible que les sondes déclenchent l'ouverture d'une connexion pour atteindre une partie du réseau. Cette hypothèse semble à exclure puisque dans un tel cas les sondes suivantes et passant par ce chemin seraient aussi retardées ; dans nos observations, les réponses pour les sauts suivants sont arrivées avec un délai normal (moins de 200 ms), ce qui indique qu'elles ont traversé sans délai le routeur incriminé et atteint les routeurs suivants. Nous pouvons aussi envisager que les sondes ou les réponses ont été retenues dans le réseau, par exemple à cause d'une boucle de routage intermittente. Les TTL des réponses indiquent qu'elles n'ont pas traversé un nombre anormalement élevé de routeur (environ 15 dans les exemples observés), ce qui infirme cette deuxième hypothèse. Enfin, il reste l'hypothèse de la surcharge temporaire de plan de contrôle du routeur incriminé. Ce plan de contrôle est responsable des calculs réalisés de manière régulière (calcul des tables de routage, gestion de l'interface, envoi des messages d'erreur). Ce travail est généralement réalisé par un processeur généraliste, contrairement au transfert des paquets qui nécessite un matériel dédié pour atteindre les débits requis par les réseaux actuels. Il est donc envisageable que ce processeur soit momentanément surchargé à cause d'une activité qui bloque tout autre calcul, ce qui retarde la prise en charge des messages d'erreur ICMP. Plusieurs observations semblent confirmer cette hypothèse : les réponses des routeurs incriminés ne sont pas systématiquement retardées. Pendant certaines périodes, les RTT sont normaux, correspondant à des périodes où le routeur est disponible. Lorsqu'elles sont retardées, les réponses arrivent dans l'ordre dans lequel les sondes ont été émises, mais avec un délai inter-réponse très atténué par rapport au délai inter-sonde : par exemple, pour un routeur observé, il n'y a qu'un délai de 13 ms entre les 3 réponses alors que les 3 sondes sont envoyées avec un délai de 50 ms entre chaque. Cela tend à montrer que les réponses ont été bloquées puis réémises en un temps très court.

Bilan : délai d'attente des réponses. Nos résultats montrent qu'une temporisation supérieure à 2 secondes dans les outils de traçage n'apporte pas d'avantage significatif. De plus le temps d'attente des réponses influe sur la durée de la mesure. Par exemple, le temps de traçage moyen d'un chemin est de 12s avec une temporisation de 2s, et passe à 22s avec une temporisation de 5s.

Temps inter-sonde (ms)	50	250	500
Sondes sans réponse	5.5%	4.4%	3.4%

Tableau 1. *Impact du temps inter-sonde sur le pourcentage des sondes sans réponse (étoiles)*

3. Échecs de génération des réponses

Les standards divergent légèrement sur le comportement à adopter lorsqu'un routeur jette un paquet avec un TTL trop petit. La RFC 792 [POS 81] (1981) indique qu'un message d'erreur *peut* être envoyé à la source. En revanche, la RFC 1812 [BAK 95] (1995) est plus claire : tout routeur IPv4 *doit* signaler le problème à la source. La réalité est plus complexe, à cause des différentes implémentations, configurations et politiques de sécurité employées par les opérateurs. Cette section énumère les raisons pour lesquelles un routeur peut décider de jeter silencieusement un paquet sans générer le message d'erreur utilisé par traceroute.

3.1. *Rate-limiting*

Les standards prévoient qu'un routeur envoie un message d'erreur pour chaque paquet jeté, mais ne tiennent pas compte des problèmes de sécurité que cela implique (déni de service, en particulier). Pour palier ce problème certains routeurs offrent la possibilité de limiter le nombre de réponses qu'ils émettent dans un intervalle de temps donné (*rate-limiting*).

La détection des routeurs ayant activé cette option est une tâche délicate et difficile à discerner des autres phénomènes comme la perte de paquets. Pour cette raison, nous avons plutôt cherché à détecter ce phénomène au niveau global, c'est à dire son impact sur le nombre total de sondes sans réponse. Pour cela nous avons réalisé l'expérience suivante. Nous traçons la route vers nos 5000 destinations en envoyant 10 sondes successives à chaque routeur. Nous répétons l'expérience pour plusieurs temps d'attente entre les sondes destinées à un routeur : 50, 250, 500 ms. Des sondes émises en rafale (temps inter-sonde faible) et destinées à un même routeur ont plus de chance de déclencher le rate-limiting, avec pour résultat une plus grande quantité d'étoiles dans la trace obtenue.

Nous avons effectué 5 passes successives pour chaque temps inter-sonde. Dans chaque trace obtenue nous éliminons tous les sauts sans réponse qui peuvent éventuellement apparaître à la fin de la route (causés par une destination muette ou un équipement de filtrage à l'entrée du réseau de la destination, voir section 3.3). Le tableau 1 présente le pourcentage de sondes sans réponse pour les trois temps inter-sondes. Il est clair qu'un temps inter-sonde court a pour conséquence un plus grand nombre d'étoiles, confirmant la présence de rate-limiting sur certains chemins me-

surés. Nous travaillons actuellement sur une méthode pour détecter précisément les routeurs responsables de ce phénomène.

3.2. *Blocage*

Un problème important de la cartographie de l'internet concerne les routeurs dits anonymes [YAO 03] car ils bloquent toute émission de message ICMP. Par conséquent ils ne révèlent jamais leur présence et apparaissent systématiquement sous la forme d'une étoile. Plusieurs raisons peuvent expliquer la présence de tels routeurs. Par exemple, une implémentation incorrecte ou ne respectant pas la RFC 1812. Ou encore des opérateurs qui ne veulent pas révéler la structure de leur réseau, et désactivent, par une règle de filtrage, l'émission de messages ICMP par leurs routeurs.

Pour mettre en évidence ces routeurs "muets", nous devons les différencier des autres phénomènes causant des étoiles. Pour cela nous nous appuyons sur le caractère permanent des non-réponses. En effet, nous considérons que les autres cas de non-réponse ne sont que temporaires, de l'ordre de la seconde. Cette hypothèse est valable pour le rate-limiting et la perte de paquets. Par conséquent, si nous sondons un routeur pendant un laps de temps suffisamment long, nous diminuons les chances de confondre une non-réponse temporaire avec un cas de non-réponse permanente.

Nous avons donc réalisé l'expérience suivante. Nous traçons le chemin vers chaque destination en émettant 10 sondes par saut, avec une pause de 500 ms entre chaque sonde. Nous sondons donc chaque routeur pendant 5 secondes. Nous éliminons les sauts sans réponses qui peuvent apparaître à la fin des chemins. A partir de cette trace unique nous générons 10 traces, pour chaque trace nous ne considérons que les n premières sondes envoyées à chaque saut, pour n variant de 1 à 10. Nous évaluons ensuite la qualité de la trace obtenue, en termes de sauts sans réponse. La figure 1 montre en ordonnées la fraction de chemins comportant au moins un saut sans aucune réponse parmi les n sondes envoyées (en abscisses). Intéressons-nous à la courbe pour w = 500 ms. On observe une nette amélioration de la qualité des chemins entre 1 et 3 sondes ; cela correspond à un intervalle d'une seconde, pendant lequel la plupart des problèmes temporaires sont détectés. L'amélioration est beaucoup moins nette au delà (23% de chemins incomplets avec 4 sondes, 21% avec 10 sondes). Ce plateau semble montrer que les cas temporaires ont été résolus, et qu'il ne reste plus que les cas permanents, donc les routeurs muets. Ceux-ci affectent donc une large portion des chemins, environ 20%.

Bilan : nombre de sondes et délai inter-sonde. La figure 1 (gauche) donne des indications sur le paramétrage du nombre de sondes et du délai inter-sonde. Pour un délai de 50 ms, l'amélioration est claire jusqu'à 7 sondes. Pour des délais supérieurs, 4 sondes suffisent. Nous travaillons actuellement sur l'évaluation du compromis entre le nombre de sondes, la durée et la qualité de la mesure.

Bilan : condition d'abandon. Certains outils, en particulier Paris traceroute, abandonnent la mesure d'un chemin après avoir rencontré un nombre paramétrable de sauts

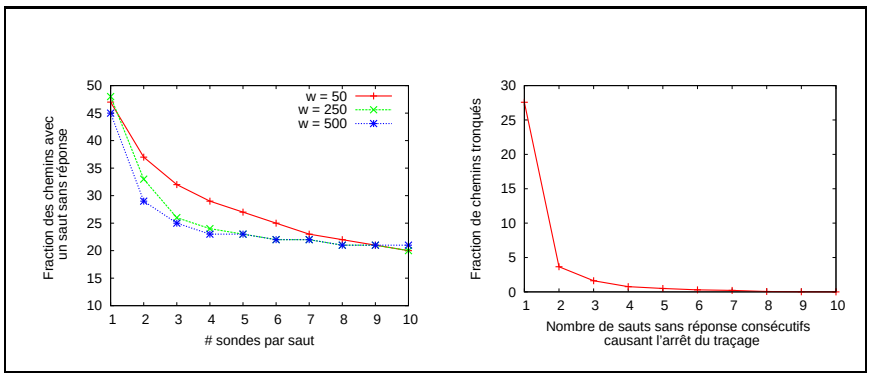


Figure 1. *A gauche : influence du nombre de sondes par saut sur la qualité du traçage, pour différents délais inter-sonde (w). A droite : influence du paramétrage de la condition d'abandon du traçage (nombre de sauts consécutifs sans réponse).*

consécutifs sans réponse, noté M. La présence de plusieurs routeurs anonymes consécutifs peut forcer l'abandon prématuré du traçage et découvrir des chemins tronqués. Pour évaluer l'impact de ce paramètre nous avons tracé les chemin vers nos 5000 destinations, en abandonnant le traçage après 10 sauts sans réponse. A partir de cette trace nous avons construit la figure 1 (droite) qui évalue le nombre de chemins tronqués (traçage abandonné trop tôt) en fonction de M. Ici encore, un compromis est nécessaire. Au delà de 3 sauts, le bénéfice n'est pas net. Notons tout de même l'existence de chemins traversant jusqu'à 9 routeurs anonymes, ce qui laisse supposer l'existence de réseaux entiers qui bloquent les sondes traceroute.

3.3. *Filtrage en périphérie*

Traceroute utilise les messages ICMP "TTL Exceeded" pour tracer la route dans le coeur du réseau. Lorsqu'il atteint sa destination, il se base sur un autre type de réponse, qui dépend du protocole utilisé : réponse d'écho ICMP, ICMP "Port Unreachable" pour UDP, SYN/ACK ou RST pour TCP. Pour des raisons de sécurité, il arrive fréquemment que l'hôte destinataire, ou encore un pare-feu à l'entrée du réseau du destinataire, bloque l'envoi de telles réponses. En conséquence, traceroute est incapable de détecter qu'il a atteint la fin du chemin, et continue à émettre des sondes avec un TTL croissant, ce qui mène à l'apparition d'une grande quantité de sauts sans réponse (sauts étoilés) en fin de chemin. Le nombre de ces sauts sans réponse en fin de chemin n'a aucune signification (contrairement aux sauts étoilés dans le coeur du réseau, ici on ne peut pas faire correspondre chaque saut à un noeud du réseau) et dépend uniquement du paramétrage de l'outil (voir section 3.2). C'est pourquoi dans l'article nous ne considérons que des traces nettoyées, c'est à dire dans lesquelles nous avons éliminé tous les sauts sans réponse en fin de chemin.

	TCP	TCP/80	UDP	ICMP
Aléatoire, répondant au ping (5000 adr.)	24	26	20	46
Aléatoire (5000 adresses)	5	5	4	6
Serveurs Web (500 adresses)	43	97	41	77
Routeurs (69000 adresses)	66	67	68	73

Tableau 2. *Fraction de destinations atteintes pour 4 types de destinations et 4 types de sondes.*

Le tableau 2 indique la fraction de destinations atteintes pour 4 types de sondes et 4 types de destinations utilisées dans des travaux précédents [AUG 06, AUG 07a]. Mis à part les serveurs Web qui répondent plutôt aux requêtes sur le port 80, il semble que les sondes ICMP produisent le plus grand nombre de réponses des destinations. UDP donne les moins bons résultats.

3.4. *Filtrage des routeurs*

Lorsqu'un filtre est installé en périphérie du réseau, il bloque l'acheminement de tout paquet interdit. Cela se traduit par l'impossibilité de tracer le chemin au delà du filtre. Cependant, il existe aussi des filtres au coeur du réseau (dans les routeurs), dont le fonctionnement est différent. Le filtre n'est pas installé au niveau du plan de routage (partie du routeur qui gère le transfert des paquets d'une interface à une autre) mais au niveau du plan de gestion. Cette partie du routeur gère le calcul des tables de routage, les messages de routage, l'interface en ligne de commande, ainsi que l'émission de messages d'erreur ICMP. Or certains routeurs appliquent un filtre sur le paquet jeté, avant de décider d'envoyer un message ICMP "TTL expiré". Cette section met en évidence ces comportements inattendus suivant deux paramètres : les numéros de port et le protocole utilisé.

3.4.1. *Numéros de port*

Le choix de la plage de ports destination utilisés par un traceroute est important car certains routeurs ne réagissent qu'à certains ports spécifiques. Pour mettre en évidence ce comportement nous lançons deux mesures simultanées, l'une utilisant le port destination 33434, l'autre le port 20000. Le but est de comparer le nombre de non-réponses obtenues en utilisant un port dans la plage du traceroute classique (33434 et suivants) et en dehors de cette plage. Nous avons répété cette mesure 10 fois, et le nombre de sauts sans réponse était systématiquement plus élevé avec le port 20000. La différence est très légère, environ 6% de non-réponses supplémentaires, ce qui montre que de tels routeurs sont très peu déployés. Nous avons isolé certains de ces routeurs (figure 2) dans nos traces. Des tests plus précis on montré qu'ils n'émettent un message d'erreur que pour les paquets dans une certaine plage de ports destination ([33434, 33534], [33400, 34400]), qui correspond à la plage utilisée par le traceroute classique.

```
13 4.68.116.145                 13 4.68.116.145
14 208.50.13.193                14 * * *
15 64.209.102.154               15 64.209.102.154
```

Figure 2. *Réponse sélective du routeur au saut 14. Il répond au port 33434 (à gauche) mais reste muet au port 20000 (à droite).*

Il s'agit d'un détournement de la fonction "TTL Exceeded", initialement prévue pour informer de la perte d'un paquet à cause d'une mauvaise configuration (boucle de routage), au profit unique du traçage de route, puisque ces routeurs ne génèrent des paquets d'erreur que pour traceroute et n'informent pas des éventuels autres problèmes (ils restent muets). Nous n'avons pas pu trouver trace d'une telle fonctionnalité dans la documentation de routeurs bien connus. L'existence de plusieurs plages de ports (nous en avons trouvé deux, mais il peut en exister bien plus, dans la mesure où nous n'avons pas réalisé de tests à plus grande échelle) pourrait indiquer que la plage est configurable par l'opérateur, à moins que cela ne traduise l'existence de différents modèles de routeurs.

Bilan : numéros de ports. L'utilisation de ports destination en dehors de la plage de traceroute cache certains routeurs, ce qui peut poser problème à Paris traceroute, qui émet des sondes avec différents numéros de ports, dans le but de détecter les répartiteurs de charge [AUG 07b]. La solution consiste soit à varier les ports destination dans la plage de traceroute, soit à varier le port source, qui ne semble filtré par aucun des routeurs rencontrés.

3.4.2. *Protocole*

De même que le port destination, le choix du protocole influe sur la génération de réponses. Nous avons collecté plusieurs traces en utilisant les protocoles UDP, ICMP et TCP, pour comparer le nombre de sauts sans réponse. Contrairement à l'expérience précédente, les résultats n'ont pas permis de conclure de manière certaine. En revanche, nous avons pu montrer l'existence du phénomène en isolant manuellement trois de ces routeurs. Deux d'entre eux répondent aux sondes UDP (le protocole par défaut du traceroute classique) mais restent silencieux aux sondes ICMP et TCP. Le troisième répond aux protocoles UDP et ICMP (proposés par traceroute classique) mais pas aux sondes TCP. Ces routeurs sont donc systématiquement cachés aux utilisateurs de tcptraceroute.

Bilan : protocole. Le choix du protocole dépend du type de destination tracée. TCP mais surtout ICMP semblent donner les meilleurs résultats en termes de destinations atteintes, mais certains routeurs seront systématiquement cachés avec les outils traçage basés sur ces protocoles, comme tcptraceroute. De plus, il ne faut pas perdre de vue que l'utilisation de TCP augmente le risque de déclencher une alerte de sécurité dans un des réseaux traversés.

3.5. *Charge*

Des phénomènes temporaires autres que le rate-limiting peuvent causer des non-réponses : pertes de paquet dû au contrôle de congestion, d'autant plus que les réponses ICMP ont une faible priorité ; non-génération d'erreur ICMP dû à la surcharge temporaire d'un routeur occupé à mettre à jour sa table de routage, par exemple. Leur caractère sporadique les rend difficiles à mettre en évidence. Nous avons néanmoins observé ces phénomènes dans nos traces, par exemple certains routeurs ayant des périodes de silence à durée variable, de l'ordre de la seconde à plusieurs heures. Nous travaillons actuellement sur des techniques pour détecter ces phénomènes et les distinguer les uns des autres.

4. Travaux antérieurs

Les outils de cartographie de l'internet ont très rapidement dû faire face au problème des non-réponses et de leur impact sur la qualité des cartes obtenues. Différentes approches ont été utilisées pour palier ce problème. Certains filtrent leur jeu de données en ignorant les chemins avec un routeur muet [PAN 98, GOV 00]. L'outil utilisé par Cheswick et. al. [CHE 00] arrête le traçage d'un chemin dès qu'il rencontre un noeud qui ne répond pas. D'autres techniques sont proposées pour construire des cartes précises en présence de noeud anonymes : création d'arcs connectant les noeuds adjacents [BRO 01], heuristiques pour fusionner des noeuds anonymes en conservant la consistance de la topologie obtenue [YAO 03]. Enfin, notons que Rocketfuel [SPR 02], qui crée des cartes des systèmes autonomes, ne fait aucune mention de ce problème.

Le rate-limiting est aussi un problème documenté. Il est autorisé par les standards [BAK 95], et Govindan et. al. [GOV 02] reconnaissent son utilisation en pratique. Leur outil en tient compte en fixant un intervalle d'une seconde entre chaque sonde émise. Savage motive l'utilisation de son outil Sting [SAV 99] en montrant que les outils basés sur ICMP sont limités à cause du filtrage de ce protocole. Paris traceroute [AUG 07a] introduit un mécanisme de retransmission des sondes sans réponse pour palier ce problème.

Le problème de l'altération des réponses ICMP lors de l'envoi de sondes TCP est décrit par Malone et al. [MAL 07]. Néanmoins, le problème d'association des réponses dans les outils de traçage n'est pas évoqué.

5. Conclusion

Cet article apporte trois contributions. Premièrement, nous énumérons et classons les causes d'apparition des étoiles dans les outils de traçage. Deuxièmement, nous proposons des mesures qui permettent de mettre en évidence l'existence de chacune de

ces causes. Finalement, nous montrons comment ces résultats peuvent nous permettre de mieux paramétrer nos outils.

L'interprétation de ces résultats n'est pas triviale, et nécessite souvent un compromis entre l'effort fourni pour le traçage (en temps et en nombre de paquets émis), et la qualité des traces obtenues. Des travaux sont en cours, pour mieux évaluer ce compromis. En particulier, nous pensons qu'une caractérisation plus fine du rate-limiting nous permettrait d'apporter des améliorations majeures à notre outil, Paris traceroute, qui nécessite l'envoi massif de sondes pour découvrir les chemins en présence de répartition de charge. Nous disposons aussi d'une plate-forme constituée de routeurs hétérogènes, sur laquelle nous menons actuellement des mesures pour mieux comprendre les mécanismes de génération de messages ICMP au sein des routeurs.

6. Bibliographie

[AUG 06] AUGUSTIN B., CUVELLIER X., ORGOGOZO B., VIGER F., FRIEDMAN T., LATAPY M., MAGNIEN C., TEIXEIRA R., « Avoiding traceroute anomalies with Paris Traceroute », *Proc. ACM SIGCOMM Internet Measurement Conference, IMC*, October 2006.

[AUG 07a] AUGUSTIN B., FRIEDMAN T., TEIXEIRA R., « Measuring Load-balanced Paths in the Internet », *Proc. ACM SIGCOMM Internet Measurement Conference, IMC*, October 2007.

[AUG 07b] AUGUSTIN B., FRIEDMAN T., TEIXEIRA R., « Multipath Tracing with Paris Traceroute », *Proc. IEEE Workshop on End-to-End Monitoring, E2EMON*, May 2007.

[BAK 95] BAKER F., « Requirements for IP Version 4 Routers », RFC1812, June 1995.

[BRO 01] BROIDO A., K CLAFFY, « Internet topology : connectivity of IP graphs », *Proc. Workshop on Scalability and Traffic Control in IP Networks, SPIE ITCOM Conference*, August 2001.

[CHE 00] CHESWICK B., BURCH H., BRANIGAN S., « Mapping and visualizing the internet », *ATEC'00 : Proceedings of the Annual Technical Conference on 2000 USENIX Annual Technical Conference*, 2000.

[CHO 04] CHOI B.-Y., MOON S., ZHANG Z.-L., K. P., DIOT C., « Analysis of Point to Point Packet Delay In an Operational Network », *Proc. IEEE Infocom*, March 2004.

[GOV 00] GOVINDAN R., TANGMUNARUNKIT H., « Heuristics for Internet Map Discovery », *Proc. IEEE Infocom*, March 2000.

[GOV 02] GOVINDAN R., PAXSON V., « Estimating Router ICMP Generation Delays », *Proc. of Passive and Active Measurement Workshop, PAM*, 2002.

[MAL 07] MALONE D., LUCKY M., « Analysis of ICMP Quotations », *Proc. of Passive and Active Measurement Workshop, PAM*, 2007.

[PAN 98] PANSIOT J., GRAD D., « On routes and multicast trees in the Internet », *ACM Computer Communication Review*, January 1998.

[POS 81] POSTEL J., « Internet Control Message Protocol », RFC792, 1981.

[SAV 99] SAVAGE S., « Sting : A TCP-based Network Measurement Tool », *USENIX Symposium on Internet Technologies and Systems*, October 1999.

[SPR 02] SPRING N., MAHAJAN R., WETHERALL D., « Measuring ISP Topologies with Rocketfuel », *Proc. ACM SIGCOMM*, August 2002.

[YAO 03] YAO B., VISWANATHAN R., CHANG F., WADDINGTON D., « Topology Inference in the Presence of Anonymous Routers », *Proc. IEEE Infocom*, April 2003.

Réseaux sans-fil ad hoc

Développement et analyse multi outils d'un protocole MAC déterministe pour un réseau de capteurs sans fil

Thierry Val — Adrien van den Bossche

LATTIS EA4155, Groupe SCSF, Université de Toulouse
IUT de Blagnac - UTM
BP60073, 1 place Georges Brassens, F-31703 Blagnac
{val, vandenbo}@iut-blagnac.fr

RÉSUMÉ. *Nous présentons dans cet article une méthodologie multi outils de développement et d'analyse de protocole utilisée pour valider une nouvelle méthode d'accès. La technologie IEEE 802.15.4 / ZigBee sert de base protocolaire à la proposition d'une couche MAC déterministe offrant un haut niveau de QdS. Ce type de WPAN peut typiquement être utilisé pour des réseaux de capteurs sans fil à fortes contraintes temporelles. Afin de valider les protocoles proposés, trois outils complémentaires et adéquats sont utilisés : les Réseaux de Petri pour la validation formelle du séquencement des trames, un simulateur spécifique pour les aspects temporels et des métrologies sur un prototypage réel à base de composants ZigBee FREESCALE pour la caractérisation fine des couches physique et liaison.*

ABSTRACT. *In this article, we present a multi-tool method for the development and the analysis of a new medium access method. IEEE 802.15.4 / ZigBee technology has been used as a basis for this new determinist MAC layer which enables a high level of QoS. This WPAN can be typically used for wireless sensor networks which require strong temporal constraints. To validate the proposed protocol, three complementary and adequate tools are used: Petri Nets for the formal validation of the algorithm, a dedicated simulator for the temporal aspects, and some measures on a real prototype based on a couple of ZigBee FREESCALE components for the hardware characterization of layers #1 and #2.*

MOTS-CLÉS : *Réseau de capteur sans fil, IEEE 802.15.4, ZigBee, Méthode d'accès au médium, Qualité de Service, Prototypage, Validation formelle, RdP, Simulation, Analyse de performance, métrologie.*

KEYWORDS: *Wireless sensor networks IEEE 802.15.4, ZigBee, Medium access control, Quality of Service, Prototype, Formal validation, RdP, Simulation, Performance analysis, metrology.*

1. Introduction

La conception de protocoles de communications est un domaine de recherche à part entière. De nombreuses méthodes et outils ont été développés et apportent des aides évidentes aux chercheurs et industriels. Suivant les compétences des acteurs et leurs origines, on utilise initialement de façon privilégiée plutôt des outils de spécification et de validation formelle, ou plutôt des outils de simulation de réseaux et de protocoles, voire du prototypage réel. Chaque technique apporte son lot d'avantages en fonction des besoins et du niveau d'abstraction utilisé. Plus les protocoles à développer et valider sont proches des couches basses, plus il semble naturel d'utiliser un niveau de granularité fin, proche du matériel et des caractéristiques intrinsèques des médiums. Ceci est d'autant plus vrai lorsque l'on parle d'ingénierie des protocoles pour des réseaux sans fil où les caractéristiques dynamiques des médiums immatériels sont prépondérantes.

Les auteurs de cet article ont été confronté à ce genre de choix d'outils et de méthodes, lors de la proposition et la validation d'une nouvelle méthode d'accès dédiée à un réseau sans fil de capteurs pour des applications à très fortes contraintes temporelles, imposant un déterminisme fort de cette couche MAC. Nous avons alors décidé d'utiliser de façon ciblée et complémentaire plusieurs techniques, adéquates, dans le développement du protocole. Cet article présente donc tout d'abord les caractéristiques du WPAN ZigBee/802.15.4 qui a servi de base à nos réflexions. Après avoir identifié ses lacunes face à notre cahier des charges, nous présentons notre proposition de nouvelle couche MAC déterministe. La plateforme de validation multi outils est ensuite détaillée en présentant l'utilisation des RdP pour l'aspect formel de la validation, la conception d'un simulateur dédié et les analyses de performances temporelles qui en découlent, et enfin, les métrologies liées à un prototypage réel sur composants ZigBee de Freescale/Motorola.

2. Présentation de la norme de communication IEEE 802.15.4/ZigBee

2.1. *Généralités*

Nos travaux se basent sur la technologie de communication sans fil courte portée IEEE 802.15.4/*ZigBee* (ZigBee Alliance, 2005) (Val *et al.*, 2008), qui propose une norme pour des communications bas débit pour des entités embarquées (réseaux de capteurs par exemple). Un émetteur/récepteur ZigBee est caractérisé par une portée de 10 à quelques centaines de mètres et un débit de 20 à 250 kbit/s. La norme prévoit l'utilisation de trois bandes de fréquence (868, 915 ou 2400 MHz). La spécification ZigBee propose une pile protocolaire propriétaire et légère, déclinable dans plusieurs versions. Elle s'appuie sur la norme IEEE 802.15.4 (IEEE, 2003) pour les couches *Physique* et *Liaison de données*. Elle propose ses propres couches supérieures (*Réseau*, etc.). ZigBee réalise de fortes économies d'énergie grâce à une optimisation des périodes de mise en veille du matériel (Freescale, 2005).

2.2. *Accès au médium radio*

La norme IEEE 802.15.4 prévoit deux modes complémentaires pour l'accès au médium radio : un mode avec contention, de type *Best Effort*, par utilisation du protocole CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) ; et un mode avec réservation de ressources par créneaux temporels garantis (GTS, *Guaranteed Time Slots*), où seul l'élément détenteur du GTS est autorisé à émettre.

Ce dernier mode permet d'entrevoir une approche de Qualité de Service garantie par un élément *coordinateur du réseau* ; cet élément est chargé de répartir les accès au médium des entités communicantes qui lui sont rattachées (on parle alors de topologie *en étoile*) dans une structure temporelle appelée *supertrame*. Pour distribuer les temps de parole, le coordinateur diffuse à la demande ou régulièrement des trames balises, ou *beacons*, permettant la synchronisation des nœuds communicants et la diffusion des informations relatives à la distribution des temps de parole. L'espace inter-balises ou supertrame comprend une période *active* où les éléments échangent les messages et une période *inactive*, optionnelle. La période active de la supertrame est divisée en 16 slots temporels de longueur égale ; le *beacon* du coordinateur occupe le début du premier slot, suivi par la période des accès avec contention, ou CAP (*Contention Access Period*), dans laquelle les accès au médium respectent le protocole CSMA/CA slotté. Si les feuilles en ont fait la demande en CSMA/CA auprès du coordinateur, les GTS occupent les derniers slots de la supertrame dans la CFP (*Contention Free Period*).

2.3. *Les faiblesses de la méthode d'accès face à notre problématique de recherche*

IEEE 802.15.4 propose donc un mécanisme de réservation de ressources permettant à quelques feuilles privilégiées de s'affranchir des collisions, principale source d'incertitudes temporelles au niveau *Liaison de données*. Nous souhaitons développer une couche MAC qui offrira un accès au médium entièrement garanti sans aucune collision potentielle. Compte tenu de ce cahier des charges arbitraire très contraignant, nous proposons de renforcer ce mécanisme par la mise en œuvre d'une couche MAC à forte QdS, comme cela peut être requis dans le cadre d'un réseau de capteurs ou, plus largement, dans le cadre de réseaux industriels sans fil (Lepage, 1991). Les améliorations proposées se justifient par plusieurs faiblesses identifiées comme telles compte tenu de nos objectifs.

– L'obtention d'une portion temporelle dédiée est conditionnée premièrement par la non saturation préalable du réseau. Sa capacité n'est pas infinie, mais la norme ne fournit aucune possibilité au coordinateur de *réserver à priori* une partie des ressources pour certains nœuds *critiques* d'un point de vue applicatif. Selon la norme IEEE 802.15.4, les premiers demandeurs sont les premiers servis, ce qui n'est pas une politique de répartition acceptable compte tenu de nos objectifs de déterminisme fort. De plus, le processus de demande, initié par l'élément désirant obtenir un GTS, génère un message envoyé au coordinateur dans la CAP, en mode *Best Effort*, donc sans aucune garantie de succès dans un temps borné.

– Si il est possible d'augmenter la longueur d'un GTS (de un à plusieurs slots) dans chaque supertrame, la norme ne prévoit pas en revanche une possibilité d'allocation moins fréquente, par exemple un GTS toutes les deux supertrames, de manière à conserver un accès dédié, mais moins fréquent ; les GTS reviennent périodiquement à la fréquence de la supertrame, fixée par le paramètre BO (*Beacon Order*), ce qui peut se révéler très consommateur de bande passante. Autrement dit, il n'est pas prévu, dans le mode sans contention, de pouvoir faire cohabiter des trafics présentant des physionomies différentes sans "gâcher" des slots.

– Si plusieurs coordinateurs cohabitent dans une même zone de portée, il y a risque de collision, même dans le cadre des GTS. En effet, la norme ne prévoit pas de mécanisme de communication entre coordinateurs ; il est alors possible que plusieurs coordinateurs voisins attribuent le même slot à l'une de leurs feuilles qui risquent alors d'entrer en collision les unes avec les autres.

Le chapitre 3 présente les principes et les améliorations que nous avons mis en place, pour fiabiliser cet accès au médium.

3. Proposition protocolaire d'une nouvelle couche MAC déterministe

3.1. *Eléments du système de communication et liens entre éléments*

Notre proposition prévoit trois catégories d'éléments communicants :

– un supercoordinateur (*PAN Coordinator*) unique. IEEE 802.15.4/ZigBee prévoient cette entité spéciale dont le rôle principal est d'attribuer des adresses. Dans notre proposition, cet élément se voit doté de nouvelles responsabilités pour la synchronisation et l'attribution des GTS. Notre solution consiste en une centralisation de toutes les demandes de réservation qui sont acheminées via les coordinateurs jusqu'au supercoordinateur qui dispose d'une connaissance exhaustive de tous les slots réservés. Cette entité centrale est à portée radio de tous les coordinateurs ; cette hypothèse est rendue peu restrictive par l'utilisation de modules IEEE 802.15.4 dotés d'amplificateurs comme le XBEEPRO (MaxStream, 2006) offrant une portée de l'ordre du kilomètre en champ libre (ce qui est largement suffisant pour grand nombre d'applications). De plus, cette centralisation permet de régler le problème du coordinateur caché. Chaque coordinateur se voit attribué un slot dédié à l'émission de sa balise, réglant ainsi les collisions de *beacons*. Nous appellerons ces « GTS dédiés aux *beacons* » : GBS (*Guaranteed Beacon Slot*),

– un coordinateur pour chaque étoile. Chaque coordinateur est à portée radio de ses entités terminales qu'il connaît à priori grâce à l'applicatif. Il peut aussi communiquer avec d'autres coordinateurs et avec le supercoordinateur (par un trafic garanti dans notre proposition). Le coordinateur relaie les demandes de GTS de ses feuilles vers le supercoordinateur,

– une ou plusieurs entités terminales, ou feuilles (capteurs / actionneurs) par étoile. Chaque feuille ne peut dialoguer qu'avec son coordinateur en CSMA/CA ou

grâce à ses GTS et PDS attribuées par son coordinateur. Un PDS (*Previously Dedicated Slot*) est un GTS attribué au préalable pour assurer une entrée déterministe dans le réseau. (cf. 3.2.4)

La topologie compte donc deux types de communication : entre le supercoordinateur et ses coordinateurs, et entre chaque coordinateur et ses feuilles

3.2. *Organisation de l'accès déterministe au médium*

3.2.1. *Notion de supertrame*

Dans notre proposition, seule la supertrame du supercoordinateur commence au slot 0 et se termine au slot 15. Les supertrames émises par les coordinateurs d'étoiles commencent à un slot *i* et se terminent à un slot *i* – 1 (modulo 16), i étant le slot du GBS. Nous appelons *superbeacon* le *beacon* émis par le supercoordinateur ; le *superbeacon* occupe chaque slot 0. Les *beacons* sont répartis entre deux *superbeacons* comme l'illustre la figure 1.

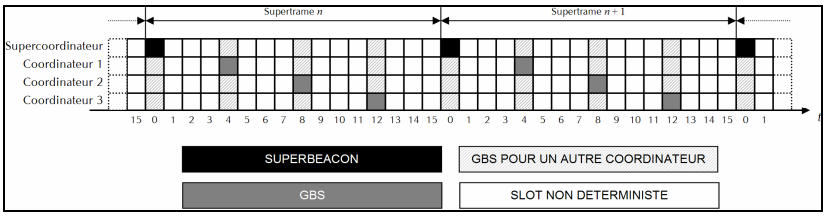


Figure 1. *Répartition des balises via les GBS dans la supertrame*

Ici, la supertrame du coordinateur 1 commence au slot 4 et se termine au slot 3 suivant. Pour que les *beacons* puissent être diffusés sans risquer d'entrer en collision avec un autre message, les slots qu'ils occupent leurs sont dédiés par le supercoordinateur qui choisit la valeur de *i* pour chaque coordinateur. Ainsi, le supercoordinateur peut répartir les émissions des *beacons* dans la supertrame, par exemple en la divisant en parts égales comme illustré par la figure 1. La frontière entre CAP et CFP est désormais supprimée ; les GTS et GBS peuvent être positionnés sur n'importe quel slot de la supertrame, en fonction des slots déjà attribués. Le choix de cet ordonnancement est laissé à l'application (regroupement pour favoriser de longues plages de veille ou, au contraire, dispersion pour favoriser une plus large distribution temporelle). En plus des slots réservés, les *beacons* doivent désormais annoncer explicitement les slots dédiés aux accès en CSMA/CA.

3.2.2. *Niveau de réservation 'n' variable*

Dans 802.15.4, les GTS alloués reviennent dans chaque supertrame. Dans notre proposition, chaque GTS peut être présent, au choix, dans toutes les supertrames, dans une supertrame sur deux, une sur quatre ou une sur huit, etc. On parlera de

plusieurs niveaux de réservation notés *n*, avec, pour le cas de notre étude la distribution *n* = 0, 1, 2 ou 3. La période P des supertrames comprenant un GTS de niveau *n* est définie selon $P = 2^n$. Ce dernier point va permettre à des trafics différents de pouvoir cohabiter au sein d'une même étoile en respectant les contraintes de QdS fixées par l'application.

3.2.3. *Mécanisme de demande de réservation du médium*

Pour pouvoir accorder un GTS à l'un de ses nœuds, un coordinateur doit d'abord envoyer une requête au supercoordinateur et obtenir une réponse positive car seul cet élément a une vision exhaustive de la répartition des slots pour les $2^{n_{MAX}}$ supertrames à venir. Les dialogues déterministes entre le supercoordinateur et les coordinateurs prennent place dans les GBS, slots dédiés par le supercoordinateur pour chacun de ses coordinateurs. De plus, les GBS, s'ils sont simplement utilisés pour la diffusion des *beacons*, constituent une perte importante de bande passante. Nous proposons ainsi d'inclure la demande de GTS directement dans les *beacons* émis par les coordinateurs.

3.2.4. *Allocations au préalable : concept de PDS*

Nos objectifs en terme d'accès déterministe au médium impliquent que toutes les opérations réalisées sur le réseau puissent être effectuées de manière déterministe. Notre proposition prévoit donc une possibilité de réservation de slots *au préalable* (PDS), c'est-à-dire avant même que certains éléments demandent leur attachement au réseau. Cette fonctionnalité est indispensable pour garantir les propriétés déterministes de notre réseau pendant toutes les phases de son fonctionnement, y compris au moment de sa création. Elle est rendue possible car le réseau considéré a une taille limitée (environ 30 nœuds par exemple). Toutes les entités potentielles peuvent être connues avant la phase de création. Le supercoordinateur peut, dès sa mise en fonction, réserver des slots pour certaines entités comme les coordinateurs ou les capteurs critiques d'un point de vue applicatif. Ces PDS peuvent également permettre un changement de coordinateur sans rupture du service d'accès au médium garanti, par exemple dans le cas d'un nœud mobile. Il est important de noter que tant que l'entité détentrice du PDS ne s'annonce pas, les PDS constituent une ressource perdue pour le système de communication. Cependant, comme les GTS et les GBS, les PDS sont attribués avec un niveau de réservation *n* en fonction de la criticité de l'élément concerné, permettant ainsi de jouer sur le rapport *période du PDS* / *ressources perdues*. Une étude théorique analysant les ressources perdues par un PDS inutilisé est détaillée dans (van den Bossche A., 2007).

3.2.5. *Accès simultanés : concept de SGTS (Simultaneous GTS)*

Dans certaines conditions de propagation, nous proposons également des GTS simultanés attribués dans deux étoiles suffisamment éloignées pour ne pas se perturber (van den Bossche A., 2007). Le protocole est le suivant : supposons un réseau constitué de deux cellules (1, 2) comportant chacune un coordinateur (C1, C2) et une feuille (F1, F2). Les deux feuilles dialoguant sans collision dans un GTS,

le coordinateur de l'autre cellule peut alors tenter de capter le signal émis et évaluer la puissance avec laquelle il reçoit la feuille de l'autre cellule. Soit PF1 la puissance reçue par C1 de sa feuille F1 et PF2 la puissance reçue de F2 ; si PF1 est supérieure à PF2 + marge, alors C1 ne sera alors pas perturbé par F2 si F1 et F2 émettent en même temps. C1 peut alors proposer un SGTS (F1, F2) à C2 qui devra procéder de la même manière. Si les deux coordinateurs arrivent à la même conclusion, le supercoordinateur peut regrouper les GTS de F1 et F2 en un même SGTS, permettant une utilisation optimisée de la bande passante totale du réseau. Le prototypage et la métrologie effectuée sur les conditions de propagation et de réception, présentés en 4.3 nous ont permis de valider ce concept et d'en déterminer les conditions optimales (marge > 10 dB).

4. Plateforme multi outils de développement et de validation protocolaire

Les nouveaux concepts présentés dans cette couche MAC originale doivent être validés. Plusieurs niveaux sont à traités : des aspects protocolaires généraux pour détecter des blocages ou des boucles dans le séquencement protocolaires des trames, des analyses de performances du point de vue temporel, mais aussi des mesures de propagation et de réception radio. Il nous semblait assez illusoire de pouvoir utiliser uniquement un seul outil. Nous avons au contraire décidé d'employer pour chaque problématique le meilleur outil, offrant la granularité suffisante et le niveau de validation de plus grand, en tentant à chaque fois de prendre en compte également la simplicité et rapidité de mise en œuvre. La figure 2 présente l'utilisation conjointe de trois outils et méthodes permettant de valider au moins à chaque fois une problématique. Avant tout travail affiné sur les analyses de performances et sur la caractérisation de la couche physique, il nous a semblé indispensable de valider formellement le séquencement protocolaire de ce nouveau protocole MAC. C'est par une modélisation par différents Réseaux de Petri (RdP), associée à des méthodes analytiques et l'utilisation de l'outil TINA (Berthomieu, 2004) développé au LAAS par le groupe OLC (Laas, 2008), que nous avons validé les différentes phases protocolaire, en prouvant le caractère borné, vivant et réinitialisable des différents RdP élaborés. Nous présenterons à titre d'exemple en 4.1. le RdP ayant permis de valider formellement l'association déterministe dans le réseau sans fil.

Il existe depuis peu plusieurs outils de simulation (OPNET, NS2…) disposant de modèles associés au réseau ZigBee. NS2 inclut des modèles de 802.15.4 (Zheng J, 2006), (Samsung, 2007) depuis la version 2.27. Ces modèles, comme ceux d'OPNET, sont généralement des contributions de recherche qui sont maintenant incluses et distribuées avec les simulateurs. Ils ont été étudiés pour traiter un ou plusieurs points particuliers mais ne modélisent pas toute la norme 802.15.4 / ZigBee (et encore moins notre couche MAC). Dans notre cas, nous souhaitions disposer d'un outil simple de simulation afin d'évaluer les aspects temporels de nos contributions spécifiques sur la couche MAC, avec l'idée de se servir des algorithmes utilisés en simulation pour faciliter le prototypage et

l'implémentation sur microcontrôleur. Il était alors bien plus rapide et efficace de concevoir un simulateur spécifique pour cela, l'idée finale étant de l'associer aux autres outils (RdP et prototype), bien plus performants et fiables pour valider les autres aspects. Nous présenterons en 4.2. les résultats de simulation obtenus à partir de notre simulateur pour caractériser la latence pour l'association déterministe par PDS. Ce simulateur nous a aussi permis d'évaluer les performances de l'accès en CSMA/CA dans la CAP, ce point là ne sera pas détaillé ici, présentant moins d'originalité dans notre contribution.

	Méthodes		
	Validation formelle	Simulation	Prototypage
Séquencement de l'algorithme (exemple de l'association déterministe)	●		
Association déterministe par PDS et latence constatée pour les coordinateurs		●	
Association déterministe par PDS et latence constatée pour les nœuds		●	
Performances de l'existant avec l'accès en CSMA/CA dans la CAP		●	
Evaluation de la sensibilité du récepteur			●
Evaluation du temps de traitement d'un paquet par la pile SMAC			●
Synchronisation des nœuds d'une même étoile – gigue quantifiée			●
Synchronisation des nœuds du réseau par cascade de beacons – gigue quantifiée			●
Capacité maximale d'un slot et évaluation du débit garanti par la méthode d'accès			●
Caractérisation de l'effet de capture et validation des SGTS			●

Figure 2. *Validation multi outils et méthodes*

Les aspects protocolaires proches de la radio sont difficilement modélisables de façon parfaite avec des outils de validation formelle et des simulateurs de réseaux. Les aspects temporels liés aux vitesses d'exécution des microcontrôleurs sont souvent omis dans les simulateurs réseaux. La caractérisation des propagations radio et des phénomènes de captures, prépondérants pour une couche MAC, est également assez délicate à obtenir avec des outils de haut niveau. Notre idée a été, au contraire, de valider ces concepts matériels par un prototypage réel et une étude du médium radio. A titre d'exemple, nous présentons en 4.3 l'étude qui nous a permis de mettre en évidence l'effet de capture et d'en déterminer les paramètres de faisabilité des SGTS. D'autres problématiques ont été analysées et validées grâce au prototype : la sensibilité du récepteur radio, l'évaluation du temps de traitement d'un paquet de niveau SMAC (*Simple MAC*, interface Freescale au plus bas de la couche MAC), la qualité de la synchronisation des nœuds via les *beacons* (intra et extra étoile), et l'évaluation des débits utiles offerts par la couche MAC, en tenant compte des caractéristiques de la couche physique (taux d'erreur bit, temps de propagation, charge processeur, temps de sérialisation pour l'échange des SDU entre les couches 1 et 2…).

4.1. *Validation formelle par RdP : cas de l'association déterministe*

Lorsqu'un capteur (fils) désire se connecter au réseau, il doit s'associer à son coordinateur (père) au sein de son étoile. Ceci va lui permettre d'émettre des données en CSMA/CA et dans ses GTS obtenus. La présence, dans la supertrame, d'un PDS pour ce capteur, permet une association déterministe sans aucun risque de collision. Le RdP, présenté en figure 3, permet de modéliser ce fonctionnement.

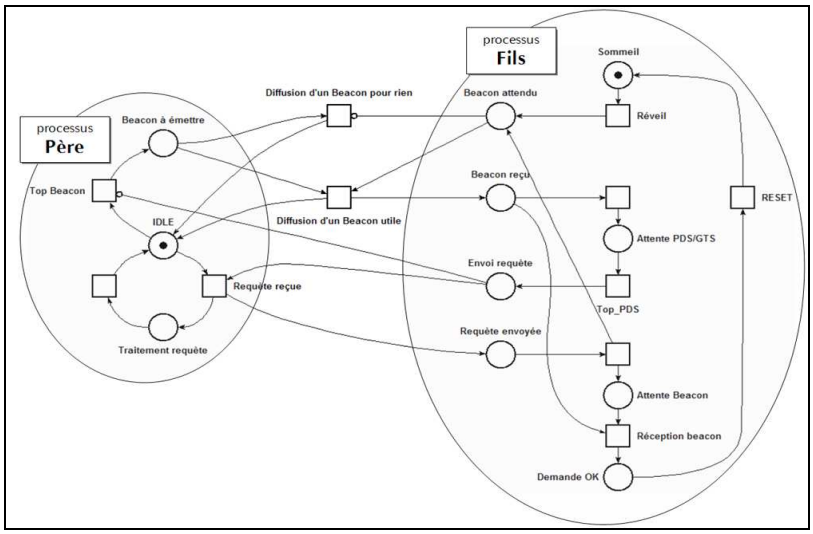


Figure 3. *RdP de l'association déterministe*

Le fils doit attendre la synchronisation émise de son père (*beacon* utile), puis son PDS, qu'il peut utiliser pour émettre des données, ou d'autres requêtes de *beacons*. Le principe est quasiment le même entre un coordinateur et le supercoordinateur. Les différents RdP, comme celui-ci présenté en exemple, ont été validé par des méthodes analytiques à l'aide de l'outil TINA. Les 3 aspects (borné, vivant et réinitialisable) ont été prouvés. Ces RdP et leur validation, ont permis de « défricher » les algorithmes de base de la méthode d'accès.

4.2. *Simulation : évaluation de la latence pour l'association par PDS*

Les différents algorithmes du protocole MAC, après avoir été validés formellement, ont servi de base à la conception d'un simulateur, écrit en langage C, puis ont été adaptés pour être prototypés. Ce simulateur s'est focalisé en particulier sur l'évaluation des latences du protocole dans la zone CFP de la supertrame. La figure 4 présente un des résultats obtenu par simulation. Nous pouvons identifier les différentes fenêtres de temps pour l'association au réseau d'un coordinateur, pour

plusieurs valeurs de n_{PDS} ($0 \leq n_{PDS} \leq 3$). Afin de se focaliser sur chaque problème un par un, ici, le médium radio est supposé parfait, c'est-à-dire sans erreur de transmission. L'axe des abscisses représente le temps nécessaire à l'association déterministe d'un coordinateur. Ce temps est découpé en intervalles car le système est temporellement discret : compte tenu du découpage en slots temporels, une demande d'association ne peut se faire qu'à certains instants. L'axe des ordonnées représente la proportion d'associations réalisées sur chaque tranche temporelle. Bien entendu, plus n_{PDS} est grand (c'est-à-dire plus la fréquence des slots réservés au préalable est faible), plus la plage temporelle sur laquelle les associations sont réparties est large.

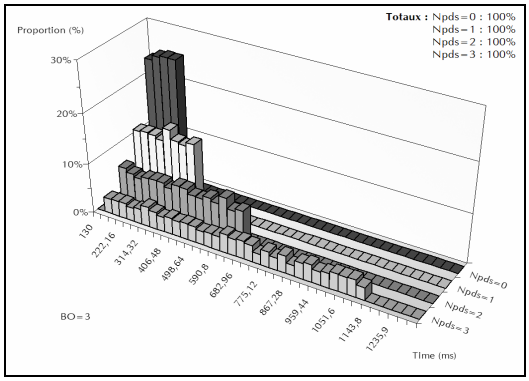


Figure 4. *Représentation des fenêtres temporelles pour l'association déterministe d'un coordinateur (pour BO = 3)*

Nous pouvons constater que la méthode d'association au réseau est déterministe puisque 100% des demandes obtiennent une réponse en un temps qui a une borne minimale et maximale. Il est possible pour un coordinateur, à partir du moment où il entre sur le réseau, d'être certain qu'il obtiendra une réponse sur son association dans un temps compris dans une fenêtre temporelle FT_{COORD} telle que $17*T_{SLOT}*2^{BO} \leq FT_{COORD} \leq 16*T_{SLOT}*2^{BO}(2^{NPDS}+1)$, ce qui est impossible à borner en CSMA/CA.

4.3. *Prototypage et métrologie : validation de la faisabilité des SGTS*

C'est par prototypage et mesures radio que nous avons validé les concepts liés au médium radio et aux caractéristiques matérielles, en particulier le concept des SGTS. Nous avons conçu et programmé au niveau MAC des cartes à base de composants FREESCALE (Freescale, 2005). Ces modules sont composés d'un circuit émetteur/récepteur 2,4 GHz spécifique (MC13192) et d'un microcontrôleur 8 bits (MC9S08GT60). A l'inverse de cartes WiFi ou Bluetooth du commerce, il est possible, avec ces composants, d'attaquer la programmation au niveau le plus bas possible, juste au dessus de la couche radio. Nous avons ainsi pu reprogrammer

complètement la couche MAC et des applications de tests et de mesures. Un des points les plus sensible à valider est le concept des SGTS. Pour cela, un prototype de métrologie de réception radio a été conçu autour de 5 nœuds : 1 supercoordinateur SC, 2 coordinateurs C1/C2, 2 feuilles N1/N2. Ni est associé à Ci. Tous les nœuds sont à portée radio, et les mesures ont été effectuées en chambre anéchoïque afin d'éliminer les bruits radio et limiter les réflexions. La figure 5 représente un résultat de mesure primordial pour prouver les conditions de faisabilité des SGTS.

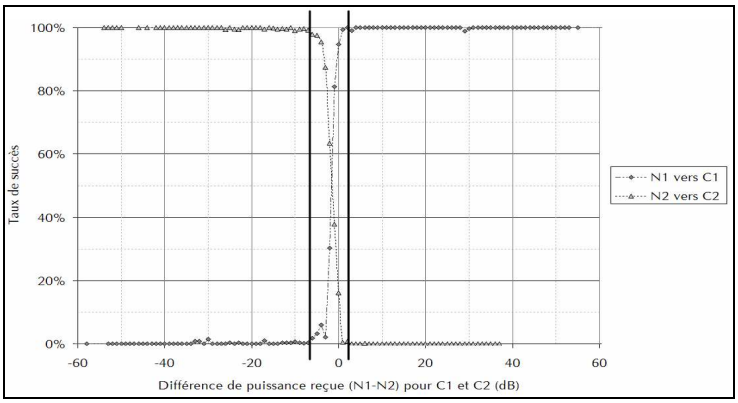


Figure 5. *Taux de trames reçues correctement en fonction du rapport de puissances reçues*

Le SC synchronise ses Ci par des *beacons* périodiques. Chaque Ci propage la synchronisation à sa feuille Ni. Chaque Ni émet alternativement une trame de données à son Ci (ceci permet d'évaluer la puissance de réception individuellement), puis les deux feuilles Ni émettent en même temps (SGTS) une trame. Ce processus est répété périodiquement en faisant varier les puissances d'émission des feuilles. La figure 5 représente le taux de trames reçues avec succès dans le SGTS en fonction de la différence de puissance reçue (N1-N2 pour C1, ou N2-N1 pour C2). Nous pouvons identifier une marge $\approx$ 10 db quasiment centrée sur 0 - avec un léger décalage du à une imperfection de synchronisation (van den Bossche, 2007) - en dehors de laquelle les SGTS sont efficaces puisque les trames de données sont reçues à pratiquement 100%. Ceci met en lumière le phénomène de capture où un récepteur se "cale" sur le premier signal reçu, même si un autre émetteur reçu légèrement plus fort débute son émission quelques instants après.

5. Conclusion

Le chapitre 4 de ce papier ne présente que quelques uns des nombreux résultats obtenus par les 3 méthodes et outils complémentaires pour valider cette nouvelle couche MAC dédiée aux réseaux de capteur sans fil. L'ensemble des résultats est

disponible sur (van den Bossche, 2007). L'utilisation conjointe des outils de modélisation et validation par RdP, d'un simulateur de réseau dédié, et d'un prototypage réel associé à des métrologies, a montré son efficacité dans ce processus d'ingénierie de protocole et de réseau sans fil. Chaque outil apporte son avantage et son niveau de granularité adéquat. Les algorithmes protocolaires ont été successivement validés formellement grâce aux RdP, évalués temporellement par simulation, puis testés en environnement réel sur la maquette. Ce prototypage final a été grandement simplifié grâce aux 2 étapes précédentes qui ont permis de corriger toutes les erreurs de conception et de codage que l'on est malheureusement amené à faire dans la conception d'un protocole de niveau MAC aussi tributaire des contraintes temporelles. Il a été bien plus simple et efficace de valider sur la maquette, les problèmes radio et ceux liés aux charges processeur implémentant la couche 1 et 2, plutôt que de se lancer dans une modélisation complexe de la couche 2.4 GHz DSSS de 802.15.4, et du matériel informatique industriel embarqué dans les nœuds ZigBee Freescale. Enfin, même si les opérations de validation ont été établies dans l'ordre présenté dans cet article (RdP, puis simulation puis prototypage), nous envisageons cependant un retour arrière qui nous permettrait d'intégrer maintenant des résultats de métrologie radio et temporelles dans le simulateur, voire même dans les modèles RdP.

Références

Berthomieu B. et *al*. « The tool TINA - Construction of Abstract State Spaces for Petri Nets and Time Petri Nets », *Int. Journal of Production Research, Vol. 42, no14*, 07/2004.

Freescale Semiconductors, MC13192 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard: Reference Manual Document, MC13192RM Rev. 1.3, 2005.

LAAS, *Le site web du groupe OLC* : http://www2.laas.fr/laas/1-4286-Groupe-OLC.php

LAN-MAN Standards Committee of the IEEE Computer Society, *802.15.4 IEEE Standard for Information technology, Part 15.4: Wireless Medium Access Control & Physical Layer. Specifications for Low-Rate Wireless Personal Area Networks*, Std 802.15.4-2003.

Lepage F., *Les Réseaux Locaux Industriels : Principes illustrés par des ex.*, Hermès, 1991.

MaxStream, Inc, IEEE 802.15.4/ZigBee XBEEPRO RF Modules datasheet, 2006.

Samsung, *NS2 simulator for 802.15.4* – http://www-ee.ccny.cuny.edu/zheng/pub

Val T., Campo E., van den Bossche A., « Technologie ZigBee / 802.15.4 : Protocoles, topologies et domaines d'application », *Techniques de l'ingénieur, traité Réseaux et Télécommunications, TEA7508*, 2008.

van den Bossche A., Proposition d'une nouvelle méthode d'accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles, Thèse, Univ. de Toulouse, 2007.

Zheng J., Lee M.-J., « A comprehensive performance study of IEEE 802.15.4 », *Sensor Network Operations*, IEEE Press, Wiley Interscience, 2006, chapitre 4, pp. 218-237.

ZigBee Alliance, *ZigBee Specification*, ZigBee Document 053474r06, version 1.0, 2005.

Partage équitable de la bande passante dans les réseaux ad hoc

Rémi Vannier* — **Isabelle Guérin-Lassous****

* *ENS Lyon / LIP - 46, allée d'Italie - 69364 Lyon Cedex*
remi.vannier@ens-lyon.fr

** *Université de Lyon / LIP - 46, allée d'Italie - 69364 Lyon Cedex*
isabelle.guerin-lassous@ens-lyon.fr

RÉSUMÉ. *Un certain nombre de travaux ont montré que le mode DCF du standard IEEE 802.11 n'est pas adapté à de nombreuses configurations de réseaux ad hoc. Dans ces situations, on constate de fortes inégalités de répartition de bande passante entre flux. Des solutions ont été proposées pour résoudre ces inégalités, soit au niveau de la couche MAC, soit en régulant la bande passante des flux à un niveau plus élevé. Ces dernières reposent généralement sur un modèle complexe de partage du medium utilisant un graphe de contention des liens, rendant ainsi les algorithmes de répartition de bande passante inutilisables en pratique. Dans cet article, nous proposons un algorithme d'allocation de bande passante distribué et dynamique reposant sur un modèle plus simple. Nous montrons par la simulation que la qualité des allocations obtenues par notre algorithme reste équivalente, tout en simplifiant grandement les algorithmes d'allocation.*[1]

ABSTRACT. *Some previous works have shown that the DCF mode within the IEEE 802.11 standard can be inappropriate for many ad hoc configurations. In some situations, the bandwidth allocation between multiple flows can be extremely unfair. Some solutions have already attempted to solve these problems, either by modifying the MAC layer, or by regulating the flow throughput at a higher level. The latter are usually based upon a complex interference model using link contention graphs, thus making the resulting bandwidth allocation algorithms hard to use practically. In this article, we propose a distributed and dynamic bandwidth allocation algorithm based on a simpler model. We show by simulation that the quality of the resulting allocations are roughly equivalent in both models, even though the algorithms are greatly simplified.*

MOTS-CLÉS : *réseaux ad hoc, équité, allocation de bande passante*

KEYWORDS: *ad hoc networks, fairness, bandwidth allocation*

1. ce travail a été financé en majeure partie par le projet Européen AEOLUS.

1. Introduction

Les réseaux *ad hoc* sont des réseaux sans fil, sans infrastructure fixe, a priori mobiles, et dans lesquels la totalité de la gestion du réseau est distribuée. De part leur nature multisaut, le partage de la bande passante dans ces réseaux est un phénomène complexe pour lequel de nombreux modèles ont été proposés.
La norme 802.11, avec son mode DCF (Distributed Coordination Function), fournit des mécanismes simples pour limiter les collisions de paquets quand les nœuds du réseau sont suffisamment proches pour s'entendre. Ces mécanismes permettent même un partage équitable de la bande passante lorsque tous les mobiles se trouvent dans une même cellule. En revanche, dans un contexte *ad hoc*, où des nœuds peuvent être hors de portée de communication, la norme 802.11 est assez inadaptée, et peut conduire à un partage très inégal de la bande passante dans des situations courantes.
Le problème des trois paires, illustré en FIG. 1, en est un exemple simple typique ([DHO 02]). Les nœuds A, C et E souhaitent émettre repectivement vers les nœuds B, D et F. Les liens en pointillé indiquent que 2 nœuds sont à portée de détection de porteuse. Les deux paires extérieures sont donc complètement indépendantes. Dans cette configuration, si l'on n'impose pas de limite d'émission aux nœuds, et qu'ils ont en permanence un paquet à émettre, le nœud central ne pourra émettre que 3% du temps [CHA 06].

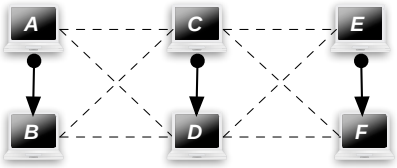


Figure 1. *Problème des 3 paires*

Il existe des solutions pour obtenir une meilleure équité dans le partage de la bande passante. Celles-ci peuvent être classées en 2 catégories :

– Des alternatives à 802.11 qui se trouvent au niveau MAC et qui ne se basent que sur des informations locales ou de voisinage, comme par exemple l'écoute du medium radio ou le fait de subir des collisions ou non. PNAV [CHA 05], MBFAIR [BEN 00] ou MadMac [RAZ 07] constituent de telles approches.

– Des solutions au-dessus de 802.11 dont le but est de réguler le débit entrant au niveau de la couche de liaison (couche 2) en fonction d'informations sur la topologie du réseau. Les travaux présentés dans [YUA 06] sont un exemple d'une telle approche.

L'inconvénient de la première catégorie est qu'il est difficile de tendre vers une équité spécifique de part le manque d'information sur la topologie. Les solutions de la 2^e classe permettent d'avoir plus de contrôle sur le partage que l'on cherche à obtenir. La solution de [YUA 06] tend, par exemple, vers une équité proportionnelle entre les

débits des flux. En revanche, ces solutions sont basées sur un modèle complexe de partage du medium et semblent, au vu de leurs simulations, trop lentes en pratique. Le modèle de partage considéré est le graphe de contention des liens du réseau dans lequel il faut très souvent identifier les cliques ou les ensembles indépendants de taille maximale.

Certaines solutions de qualité de service nécessitent aussi d'utiliser des modèles de partage du médium afin de pouvoir évaluer les ressources libres par exemple. Le protocole BRuIT [CHA 03] se base sur un modèle de partage plus simple appelé modèle par nœuds. Un tel modèle semble aussi mener à des solutions d'allocations de bande passante plus rapides.

Le but de ce travail est de regarder dans quelle mesure un modèle simple de partage du médium peut donner lieu à des allocations de bande passante efficaces. Nous cherchons à montrer qu'une imprécision dans la représentation du partage du médium radio est finalement peu importante au regard du gain obtenu, en termes de simplicité notamment, pour appliquer les algorithmes de partage. Dans cet article, nous présentons un algorithme d'allocation *équitable* de bande passante entre différents flux basé sur un modèle de partage par nœuds. Nous avons aussi comparé notre algorithme à un autre algorithme de partage similaire basé sur le modèle du graphe de contention des liens [YUA 06]. Cet algorithme est adapté aux réseaux *ad hoc*, puisqu'il est décentralisé et adaptatif, qualités essentielles dans un tel contexte. Il ne s'agit pas en revanche d'un algorithme de routage et ne remplace pas non plus la norme 802.11. L'algorithme suppose que les routes empruntées par les flux ont déjà été choisies, et adapte les débits pour résoudre les problèmes liés à la couche MAC de la norme 802.11.

Nous commencerons par examiner les difficultés liées à la modélisation des interactions dans les réseaux ad hoc dans la section 2. Nous y décrivons le modèle simplifié que nous avons choisi. La section 3 est consacrée à l'algorithme distribué de partage équitable de bande passante. En section 4, nous présentons des résultats de simulation.

2. Modélisation d'un réseau ad hoc

2.1. *Interactions*

Dans un réseau ad hoc utilisant la norme 802.11, on constate deux types d'interactions suivant qu'elles se situent au niveau du récepteur d'une transmission ou au niveau de l'émetteur :

– Au niveau du récepteur d'une transmission, des collisions peuvent se produire si un autre nœud se trouvant suffisamment proche de celui-ci émet simultanément. Pour que le récepteur puisse recevoir correctement les paquets, il faut que le rapport signal sur bruit soit supérieur à un certain seuil. Ce seuil dépend du matériel utilisé, et croît avec le débit.

– La norme 802.11 définit un seuil de puissance au-dessus duquel un nœud considère que le medium est occupé et ne peut émettre. Ce seuil définit une zone autour

d'un nœud émetteur appelée *zone de détection de porteuse* qui est approximativement deux fois plus grande que la zone de communication [DHO 03].

Dans les paragraphes qui suivent, nous présentons différentes manières de choisir les équations reflétant ces deux contraintes.

2.2. *Modèles par liens*

Ces différentes approches sont présentées dans [GUP 07]. Toutes reposent sur le *graphe de contention des liens*, i.e. un graphe dans lequel les nœuds sont les couples de nœuds *physiques* à portée de communication - les liens du réseau initial - et les arêtes symbolisent les possibilités de contention entre les liens du réseau initial.

Une première méthode consiste à identifier les *ensembles indépendants maximaux* du graphe de contention, i.e. les ensembles maximaux de nœuds n'ayant pas d'arêtes entre eux. Appelons les $I_1, I_2, \ldots I_z$. Soit n le nombre de nœuds dans le graphe de contention (et donc le nombre de liens dans le réseau initial). Alors, les débits $x_1, x_2, \ldots x_n$ des différents liens sont faisables *si et seulement si* il existe z poids non négatifs, $\lambda_1, \lambda_2, \ldots \lambda_z$ tels que $\sum_{i=1}^{z} \lambda_i \leq 1$ et :

$$\forall j \quad x_j \leq C \sum_{i/j \in I_i} \lambda_i \qquad [1]$$

où C est la capacité du canal radio. Ce système d'équations est nécessaire et suffisant pour garantir qu'il existe un ordonnancement des communications dans le réseau tel que 2 liens voisins dans le graphe de contention ne soient pas utilisés au même instant. Cependant, ces équations peuvent s'avérer difficiles à résoudre dans un contexte distribué tel qu'un réseau ad hoc. En particulier, le calcul des ensembles indépendants est un problème dont la complexité croît exponentiellement avec le nombre de nœuds.

Une deuxième méthode consiste à calculer les cliques maximales du graphe de contention. Une clique d'un graphe G est un sous-graphe de G dont les nœuds sont tous voisins directs (i.e. un sous-graphe induit complet de G). Une clique maximale est une clique qui n'est contenue dans aucune autre clique. Si Q est l'ensemble des cliques maximales, les contraintes sur les cliques s'écrivent :

$$\forall q \in Q \qquad \sum_{i \in q} x_i \leq C \qquad [2]$$

L'avantage de ces contraintes, dans un cadre ad hoc, est que ce système d'équations peut être résolu de manière distribuée par des méthodes d'optimisation Lagrangienne, car chaque équation fait intervenir des informations relativement localisées.

Les auteurs de [YUA 06] proposent une implémentation d'un algorithme distribué de répartition de bande passante basé sur ce deuxième modèle. Afin de simplifier la construction du graphe de contention, le voisinage à deux sauts est utilisé pour modéliser la zone d'interférence autour du récepteur *et* la zone de détection de porteuse. Cela

signifie que deux liens sont voisins dans le graphe de contention si l'un ou l'autre de leurs émetteurs/récepteurs ont un voisin commun dans leur zone de communication. Ensuite, à l'aide de méthodes d'optimisation Lagrangienne, les auteurs proposent un algorithme distribué pour établir une allocation équitable entre les flux du réseau ad hoc, tout en vérifiant les contraintes [2].
Cette approche présente plusieurs inconvénients. D'abord, le calcul des cliques maximales reste un problème difficile à résoudre de manière distribuée. De plus, leurs simulations montrent que la convergence de l'algorithme de partage est très lente et nécessite de nombreux échanges de messages. Les auteurs de [GUP 07] montrent qu'on peut simplifier le calcul des cliques maximales en profitant de la structure du graphe de contention des liens : en effet, en admettant que le milieu est homogène, et en assimilant un lien radio à un point situé entre les deux nœuds, le graphe de contention a une structure de Unit Disk Graph, sur lequel il existe des algorithmes simples de calcul des cliques maximales.
D'autre part, le système d'équations [2] est nécessaire, mais insuffisant pour garantir l'existence d'un ordonnancement des communications. Il n'est suffisant que dans le cas d'un graphe de contention parfait[1]. Compte-tenu enfin de l'approximation faite en restreignant la zone d'interférences au voisinage à deux sauts, ou d'éventuelles approximations des cliques du graphe de contention, les contraintes ne sont malgré tout pas complètement précises.

Cette constation nous a amenés à réfléchir à un modèle plus simple qui, s'il n'est pas idéal, pourrait aussi garantir un partage de la bande passante suffisamment équitable, tout en étant plus rapide à mettre en œuvre.

2.3. *Un modèle par nœuds*

Le protocole BRuIT [CHA 03] [2] introduit un modèle simple appelé modèle par nœuds pour faire du contrôle d'admission dans les réseaux ad hoc. Nous proposons d'utiliser ce modèle pour un partage *équitable* de la bande passante.

Dans ce modèle, le graphe de contention est construit sur le graphe de communication. Les nœuds de ce graphe sont donc les nœuds physiques, et non les liens, par opposition aux approches citées ci-dessus. Nous choisissons, comme dans BRuIT, un modèle de partage à deux sauts. Il y a donc une arête dans le graphe de contention pour chaque couple de nœuds distants de moins de 2 sauts radio. Ce choix modélise imparfaitement le partage du médium, mais nous avons choisi les contraintes de sorte à tenir compte de cette approximation.

Un ensemble de flux Φ traverse ce réseau. Chaque flux f de débit ϕ_f traverse un ensemble de nœuds qui participent à la transmission de ce flux. On définit pour tout nœud n l'ensemble $F(n)$ des flux utilisant le nœud n, i.e. tout flux pour lequel le nœud

1. un graphe est parfait si tous ses sous-graphes induits ont un nombre chromatique égal à la taille de la plus grande clique dans ce sous-graphe.
2. BRuIT : Bandwidth Reservation under InTerference

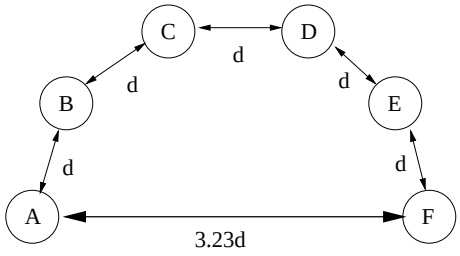


Figure 2. *Présence de zones vides de nœuds*

n joue le rôle d'émetteur et pas seulement de récepteur. Par exemple, un flux faisant le chemin $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ n'apparaîtra que dans $F(1)$, $F(2)$ et $F(3)$. Le débit du nœud n est donc :

$$x_n = \sum_{f \in F(n)} \phi_f \qquad [3]$$

Nous posons les contraintes suivantes :

$$\forall n \in N \sum_{i \in V_2(n)} x_i \leq C$$

où $V_2(n)$ est le voisinage à 2 sauts du nœud n et C est la capacité du medium radio. Ces contraintes sont les *row constraints* étudiées dans [GUP 07] pour un graphe de contention où, dans ce modèle par nœuds, l'on ne prendrait en compte que les nœuds émetteurs. D'après [GUP 07], ce système d'équations est *suffisant*, mais non *nécessaire* pour qu'il existe un ordonnancement faisable des communications tel que deux nœuds situés à moins de deux sauts n'émettent pas au même instant (interactions à l'émission). Ce modèle peut sembler surcontraint car il empêche des nœuds à une distance de quatre sauts d'émettre plus que la capacité du médium radio même si ces nœuds sont indépendants. Néanmoins, compte-tenu des interférences au récepteur, il peut être intéressant d'empêcher deux nœuds situés à quatre sauts de distance d'émettre simultanément : par exemple, si on considère le schéma de communications suivant $A \rightarrow B - C - D - E \rightarrow F$, où tous les nœuds sont alignés, et à une distance légèrement supérieure à la moitié de la distance de communication. Alors la distance entre E et B, éloignés de 3 sauts, pourrait ne pas être suffisante pour garantir que les émissions de E ne provoquent pas d'interférence au récepteur B.

Sur des simulations où les nœuds du réseau sont placés aléatoirement sur le plan, on rencontre régulièrement des situations telles que celle représentée sur la figure 2. Dans ce type de situations, bien que le réseau soit connexe, des nœuds situés à 5 sauts de distance ou plus peuvent être à portée de détection de porteuse.

3. Méthode de résolution et Algorithme

Le problème de maximisation globale des débits des flux peut alors s'écrire :

$$\text{MAXIMISER :} \qquad \prod_{f \in \Phi} \phi_f$$

$$\text{SOUS LES CONTRAINTES :} \quad \forall n \in N \sum_{i \in V_2(n)} x_i \leq C$$

où :

– C est la capacité du medium radio au niveau du nœud n.[3]

– $\prod_{f \in \Phi} \phi_f$ exprime le fait que l'on souhaite certes maximiser la bande passante allouée aux flux, mais aussi garantir une *équité proportionnelle* entre les flux.

L'optimisation Lagrangienne est une méthode couramment utilisée pour résoudre des problèmes d'optimisation linéaire avec contraintes de manière distribuée ([KEL 98], [YUA 06]). Pour ce problème, elle est particulièrement adaptée, car elle permet de transformer un problème faisant intervenir des variables globales (les débits de tous les flux traversant le réseau) en un problème d'optimisation dual qui lui n'utilise que des informations locales. Ce nouveau problème peut alors être résolu de manière décentralisée par une méthode de gradient distribué. [4]

L'algorithme de résolution obtenu par la méthode Lagrangienne est similaire à un jeu d'enchères ou de négociation de prix entre les flux et les nœuds. A chaque nœud n est associé un *tarif* λ_n dont il est responsable. Ce *tarif* représente le coût pour un flux pour utiliser la bande passante de ce nœud, que ce soit parce que ce flux est émis par le nœud n ou parce que des interférences produites par un autre nœud gênent le nœud n. Commence alors un processus itératif. A chaque pas s de l'algorithme :

– Les nœuds calculent de manière distribuée et pour chaque flux f le prix total que celui-ci devra payer pour emprunter cette route. Ce *prix* sera noté $\Pi_f^{(s)}$. Si r_{nf} est le nombre de nœuds émetteurs diffusant le flux f dans le voisinage de n, alors :

$$\Pi_f^{(s)} = \sum_{n \in N} r_{nf} \times \lambda_n^{(s)} \qquad [4]$$

– Le *prix* d'un flux est alors diffusé à l'ensemble des nœuds concernés par ce flux.

– Les nœuds réévaluent alors leur tarif en fonction de ces prix. Un nœud beaucoup utilisé va augmenter son tarif, alors qu'un nœud non saturé verra son tarif descendre à 0. A chaque pas, chaque nœud recalcule son *tarif* selon la formule :

$$\lambda_n^{(s+1)} = \lambda_n^{(s)} + \sigma \times \left(C - \sum_{f \in \Phi} \frac{r_{nf}}{\Pi_f^{(s)}}\right) \qquad [5]$$

3. Il est aussi possible de considérer la capacité restante au niveau du nœud si on suppose que des flux, comme des flux QoS, occupent déjà le réseau.
4. Pour plus de détails sur l'optimisation Lagrangienne, on peut se référer à [KEL 98] ou, pour une théorie plus complète, à [BER 82].

Figure 3. *Modélisation du problème des 3 paires*

En termes de communications, un pas correspond donc pour chacun des nœuds émetteurs à une collecte d'informations et à une diffusion dans le voisinage à 2 sauts.
On peut montrer qu'après quelques itérations de l'algorithme, la valeur unique vers laquelle convergent les prix des flux est l'inverse du débit auquel ils peuvent émettre. σ est un paramètre de l'algorithme qui régit la vitesse de convergence de l'algorithme. Une valeur de σ trop faible ralentit la vitesse de convergence, alors qu'une valeur trop forte peut conduire l'algorithme à diverger.

4. Simulations

4.1. *Cas des 3 paires*

Revenons maintenant sur le problème des 3 paires évoqué en introduction. Les 3 paires sont hors de portée de communication, et ne peuvent donc échanger de messages. C'est la présence de nœuds intermédiaires qui permet à l'algorithme de fonctionner.

Nous avons simulé une telle situation en deux temps. Dans un premier temps, nous avons calculé les débits associés à chacune des paires à l'aide de notre algorithme. Ensuite, à l'aide du simulateur *NS2* [NS-], nous avons simulé les débits recommandés par notre algorithme.
La figure 3 montre les 3 paires 0-1, 2-3 (paire centrale), 4-5, et 4 nœuds intermédiaires, de sorte que 2 paires consécutives se trouvent à une distance de 3 sauts, et à portée de détection de porteuse. Les paires extrêmes sont indépendantes. Chaque paire cherche à faire passer un flux UDP à débit constant. La bande passante du medium radio est fixée arbitrairement à 1Mb/s, et la taille des paquets à 1KB. La couche MAC est celle de l'implémentation de 802.11 par *NS-2*.
La figure 4 montre l'évolution des débits recommandés pour les 3 flux au fur et à mesure des itérations de l'algorithme (en valeur normalisée par rapport à la capacité du medium radio). On y voit que la convergence est atteinte en quelques pas, ce qui laisse envisager la possibilité d'un protocole réactif dont les limites de bande passante seraient recalculées en permanence par notre algorithme. Le tableau 5 montre quelques résultats sur la vitesse de convergence en nombre de pas. Le critère d'arrêt de l'algo-

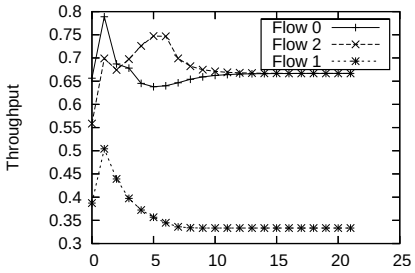


Figure 4. *Convergence de l'algorithme*

Marge d'erreur normalisée	10^{-3}	10^{-5}
Modèle par nœuds	7	11
Modèle par cliques de liens	5	10

Figure 5. *Vitesse de convergence (nombre de pas moyen) - 3 paires*

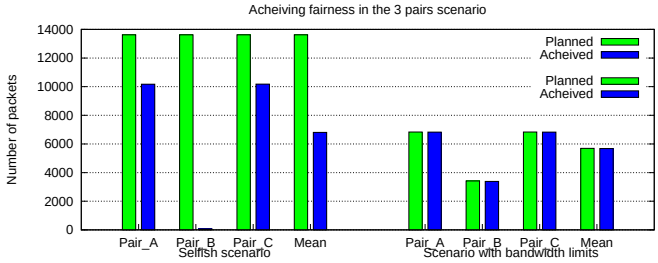


Figure 6. *Débits prévus/effectivement reçus pour les scénarios sans et avec limites de bande passante (simulation des 3 paires)*

rithme est la distance Euclidienne a la solution[5]. Les deux algorithmes, obtenus par la même méthode, mais avec des équations de contraintes différentes, sont très semblables en termes de coûts de communication par pas. Sur cette simulation, nous avons optimisé le paramètre σ de l'équation [5] afin de maximiser la vitesse de convergence dans chacun des 2 modèles, la valeur de σ optimale étant différente suivant le modèle choisi. On peut ainsi comparer l'algorithme basé sur le modèle par nœuds avec celui basé sur le modèle par cliques.

La figure 6 montre l'intérêt d'imposer des limites de bande passante pour garantir un partage équitable du medium. Pour chaque paire, les 2 barres représentent les débits que l'on impose aux paires, et ce qui a effectivement pu être envoyé. Dans le cas de gauche, sans limite, la paire centrale n'accède presque jamais au médium. En termes de débit total, cette allocation est meilleure que l'allocation de droite, mais cette dernière garantit une *équité proportionnelle* entre les flux. Cet exemple montre que l'équité proportionnelle est un compromis entre maximisation de l'utilisation du réseau et égalité parfaite.

5. si l'on note x^* le vecteur contenant les valeurs optimales des débits des flux normalisés par rapport à la capacité du medium, alors la distance entre x et x^* est $\|x^* - x\|_2$

	Paquets reçus / prévus	Pertes
Modèle par nœuds	2391 / 2478	3.54%
Modèle par liens	2440 / 2520	3.16%
Sans limite de b.p.	3536 / 8200	56.9%

Figure 7. *Moyennes du nombre de paquets réellement reçus/prévus par le modèle*

4.2. *Comparaison des deux modèles sur un cas plus complexe*

4.2.1. *Simulation*

Afin de montrer la validité du modèle par nœuds, nous l'avons comparé au modèle par liens utilisant des contraintes sur les cliques maximales du graphe de contention. Ce dernier a été construit en utilisant le modèle d'interférence à 2 sauts. Nous avons effectué une série de simulations sur un réseau de 49 nœuds disposés sur une grille 7×7 où la distance entre deux nœuds adjacents est de $200m$. Sous *NS-2*, la portée de détection de porteuse par défaut est de $550m$ et la distance de communication est de $250m$. Ainsi, certains couples de nœuds à une distance de 3 sauts sont susceptibles de se gêner. Dans chacune des expériences, nous avons choisi aléatoirement 8 liens par lesquels passent des flux en compétition. Nous avons calculé séparément les allocations optimales - au sens de l'équité proportionnelle - pour chacun des deux modèles. Nous avons ensuite simulé ces allocations sous *NS-2*, et observé la qualité des solutions obtenues par chacun des 2 modèles.

4.2.2. *Qualité des solutions*

Comme pour la simulation précédente, nous comparons le nombre de paquets envoyés à la couche MAC de l'émetteur (i.e. le débit prévu par le modèle) au nombre de paquets reçus. Dans le tableau 7, nous donnons la moyenne sur l'ensemble des simulations du nombre total de paquets envoyés par les 8 flux. Nous ajoutons pour référence les résultats que l'on obtient si l'on ne fixe pas de limite de bande passante aux nœuds.
Cette simulation[6] montre que les deux modèles aboutissent à des allocations équivalentes en termes de débit total. Pour avoir une comparaison en termes d'équité, il faut regarder la fonction d'utilité $\prod_{f \in \Phi} \phi_f$ (fonction d'utilité représentant l'équité proportionnelle). D'un exemple à l'autre, cette valeur peut être très différente entre les 2 modèles, mais en moyenne géométrique sur un grand nombre de simulations, la différence entre les 2 modèles est inférieure à 1%.

Nous avons choisi un exemple de topologie très simple, afin de pouvoir effectivement comparer les 2 modèles. Sur des graphes aléatoires, qui ne sont donc pas nécessairement connexes ou présentent des espaces vides de nœuds, il est clair qu'aucun

6. ainsi que d'autres expériences que nous avons menées en faisant varier la densité, la taille du graphe, ou en plaçant les nœuds de manière aléatoire.

Marge d'erreur normalisée	10^{-3}	10^{-4}	10^{-5}
Modèle par nœuds	46	95	115
Modèle par cliques de liens	71	90	177

Figure 8. *Etude de la vitesse de convergence - Nombre de pas moyen pour atteindre la solution en fonction de la marge d'erreur (20 simulations)*

modèle utilisant des informations de voisinage ne pourra déterminer avec précision les nœuds se trouvant dans la zone de détection de porteuse.

4.2.3. *Convergence*

Le temps de convergence des deux algorithmes induits par ces modèles est semblable. Nous constatons cependant que notre algorithme est plus adapté pour des graphes de grande taille. En effet, la vitesse de convergence des 2 algorithmes dépend du nombre de contraintes du modèle. Dans le modèle par nœuds, il y a autant de contraintes que de nœuds dans le réseau, alors que dans le modèle par liens, c'est le nombre de cliques dans le graphe de contention qui détermine le nombre de contraintes, et ce dernier augmente exponentiellement avec la taille du graphe. La figure 8 montre le nombre moyen de pas nécessaires pour atteindre la solution avec une marge d'erreur normalisée de 10^{-3}, 10^{-4}, et 10^{-5} établi sur 3 séries de 20 simulations. La convergence de l'algorithme est linéaire (i.e. la distance à la solution décroît selon une suite géométrique).

5. Conclusion

Garantir une répartition équitable de la bande passante est un problème essentiel des réseaux ad hoc que la norme 802.11 ne résout pas. Il n'existe actuellement pas de modèle à la fois simple et exact pour représenter les interactions entre les nœuds d'un tel réseau. Pour parvenir à résoudre ce problème de manière distribuée, les modèles usuels [GUP 07] [YUA 06] font de nombreuses approximations, telles que l'approximation de la zone de détection de porteuse par le voisinage à deux sauts, ou l'usage de contraintes nécessaires, mais non suffisantes. Ils conservent néanmoins un modèle de partage complexe à mettre en œuvre. Nous avons montré sur quelques scénarios qu'il est possible de choisir un modèle simple basé sur des contraintes sur les nœuds émetteurs. Celui-ci est aussi précis que le modèle usuel utilisant le graphe de contention des liens, mais simplifie grandement les algorithmes de partage, en évitant, par exemple, de calculer les cliques maximales du graphe de contention.
Notre algorithme calcule de manière distribuée des limites de bande passante sur les nœuds émetteurs en n'utilisant que des informations locales, à l'aide de méthodes d'optimisation Lagrangienne. Sur des graphes de petite taille (moins de 50 nœuds), l'algorithme converge rapidement vers une allocation équitable, ce qui laisse envisager la possibilité d'un protocole réactif aux demandes des utilisateurs du réseau, à la mobilité, ou aux pannes.

Pour la suite de ce travail, nous allons étudier plus de configurations pour affiner notre comparaison entre les deux modèles. Ensuite, nous comptons transformer notre algorithme de partage au sein d'un protocole réseau qui sera ensuite implanté sous le simulateur *NS-2* afin de l'évaluer.

6. Bibliographie

[BEN 00] BENSAOU B., WANG Y., KO C. C., « Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks », *ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Boston, Massachusetts, 2000, p. 99-106.

[BER 82] BERTSEKAS D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, Inc., 1982.

[CHA 03] CHAUDET C., GUÉRIN LASSOUS I., « BRuIT : Bandwidth Reservation under InTerferences influence », rapport, 2003, CiteSeer [http ://cs1.ist.psu.edu/cgi-bin/oai.cgi] (United States).

[CHA 04] CHAUDET C., « Autour de la réservation de bande passante dans les réseaux ad hoc », PhD thesis, INSA de Lyon, INRIA Rhône Alpes, France, September 2004.

[CHA 05] CHAUDET C., CHELIUS G., MEUNIER H., SIMPLOT-RYL D., « Adaptive Probabilistic NAV to Increase Fairness in Ad Hoc 802.11 Mac Layer », *Mediterranean Ad Hoc Networking Workshop (MedHocNet)*, 2005.

[CHA 06] CHAUDET C., GUÉRIN LASSOUS I., « Extended IFS and asynchronism influence on IEEE 802.11b medium access equity », rapport, 2006, INRIA a CCSD electronic archive server based on P.A.O.L [http ://hal.inria.fr/oai/oai.php] (France).

[DHO 02] DHOUTAUT D., GUÉRIN-LASSOUS I., « Impact of Heavy Traffic Beyond Communication Range in Multi-Hops Ad Hoc Networks », *Third International Network Conference*, Plymouth, Angleterre, 2002.

[DHO 03] DHOUTAUT D., GUÉRIN-LASSOUS I., « Experiments with 802.11b in ad hoc configurations », *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Beijing, China, September 2003, IEEE Press, p. 1618-1622.

[GUP 07] GUPTA R., MUSACCHIO J., WALRAND J., « Sufficient rate constraints for QoS flows in ad-hoc networks », *Ad Hoc Netw.*, vol. 5, n° 4, 2007, p. 429–443, Elsevier Science Publishers B. V.

[KEL 98] KELLY F., MAULLOO A., TAN D., « Rate control in communication networks : shadow prices, proportional fairness and stability », *Journal of the Operational Research Society*, vol. 49, 1998.

[NS-] NS-2, « The Network Simulator », http ://www.isi.edu/nsnam/ns/.

[RAZ 07] RAZAFINDRALAMBO T., GUÉRIN-LASSOUS I., « Increasing Fairness and Efficiency using the MadMac Protocol in Ad Hoc Networks », *Ad Hoc Networks Journal, Elsevier Ed.*, vol. doi :10.1016/j.adhoc.2007.03.003, 2007.

[YUA 06] YUAN X., LI B., KLARA N., « Optimal Resource Allocation in Wireless Ad Hoc Networks : A Price-Based Approach », *IEEE Transactions on Mobile Computing*, vol. 5, n° 4, 2006, p. 347–364, IEEE Educational Activities Department, Member-Yuan Xue and Senior Member-Baochun Li and Member-Klara Nahrstedt.

Expérimentations des communications inter-véhicules

Sofiane Khalfallah [*] — **Moez Jerbi** [**] — **Mohamed Oussama Cherif** [**] — **Sidi-Mohammed Senouci** [**] **et Bertrand Ducourthial** [*]

Lab. Heudiasyc UMR CNRS 6599 [*]
Université de Technologie de Compiègne
Compiègne, France
[*] *@utc.fr*

Orange Labs [**]
2 Avenue Pierre Marzin
22307 Lannion Cedex, France
[**] *@orange-ftgroup.com*

RÉSUMÉ. *Actuellement, les systèmes de transport intelligent (ITS) sont de plus en plus étudiés aussi bien dans la communauté de recherche que dans les industries (automobiles, télécommunications, etc.). En effet, la mise en place d'applications ITS peut à court terme améliorer la sécurité routière, augmenter efficacement l'utilisation des routes, réduire les congestions et les embouteillages, limiter l'impact des véhicules sur l'environnement, etc. Cependant, plusieurs problématiques doivent être résolues avant de pouvoir concevoir de telles applications, parmi lesquelles les communications inter-véhicules ont une place privilégiée. Dans cet article, nous nous intéressons aux performances que peuvent offrir les réseaux de véhicules multisauts, et ce dans différents scénarios, et à différents niveaux. Dans un premier temps, nous évaluons les performances du protocole 802.11 utilisé avec un routage IP statique. Dans un second temps, nous évaluons les performances d'un protocole de routage dédié aux VANETs et développé en mode utilisateur dans une architecture adaptée aux réseaux dynamiques. Notre démarche est expérimentale et deux plates-formes de tests ont été développées afin d'expérimenter sur route jusqu'à six véhicules. Ces premiers résultats donnent une idée de ce que les futures applications pour réseaux de véhicules disposeront en terme de délai ou de débit.*

MOTS-CLÉS : *réseaux de véhicules, VANET, réseaux ad hoc dynamiques, ITS, IEEE 802.11, plates-formes expérimentales, évaluation de performances.*

1. Introduction

Les communications inter-véhicules constituent un domaine de recherche récent, qui intéresse de plus en plus la communauté scientifique, les constructeurs automobiles et les opérateurs des télécommunications. En effet, les systèmes de communication inter-véhicules (IVC) peuvent être utilisés pour mettre en place plusieurs types d'applications. Ces applications sont connues sous le nom de systèmes de transport intelligents (ITS). Ces systèmes permettent d'améliorer les moyens de transport que se soit en terme de sécurité routière, de mobilité, d'impact sur l'environnement, de productivité, *etc.* Les applications ITS peuvent être à la fois mises en place dans l'infrastructure des routes et dans les véhicules eux-mêmes. Ce type de fonctionnement confère aux applications ITS trois types de déploiements : (i) véhicule-vers-véhicule (V2V), (ii) infrastructure-vers-véhicule (I2V) et véhicule-vers-infrastructure (V2I).

Dans [DUC 07a], les applications ITS, sont classées en quatre catégories : (i) applications orientées infrastructures afin d'optimiser leur gestion (gestion des autoroutes, transport de marchandises, organisation des urgences *etc.*), (ii) applications orientées véhicules afin d'améliorer la sécurité routière (alertes accidents, prévention des carambolages, conduite assistée *etc.*), (iii) applications orientées conducteurs afin de faciliter l'usage de la route (signalisation dynamique pour les embouteillages ou les travaux, estimation temps réelle du meilleur trajet, *etc.*), (iv) applications orientées

passagers pour offrir de nouveaux services à bord de la voiture (accès internet, jeux distribués, messagerie instantanée, informations touristiques *etc.*).

Plusieurs problématiques doivent être résolues avant de pouvoir concevoir de telles applications. En effet, les applications ITS exigent dans la majorité des cas des communications robustes, une certaine qualité de service ou des communications temps réelles. Or, ceci contraste avec les réseaux de véhicules qui présentent des inconvénients majeurs dus à leur forte dynamique (perte fréquente de connectivité, communication non fiable, durées de communications non bornée *etc.*). Différents travaux de recherche ont été proposés récemment pour faire face à certaines de ces problématiques [ABD 07, DUC 07a]. De nombreux protocoles de routage, de localisation ou d'adressage ont été étudiés [KHA 07b]. Des projets ont défini des architectures logicielles pour réaliser les communications véhicules-véhicules ou véhicules-infrastructures. Dans la plupart des cas, les protocoles ont été évalués par le biais de simulations [KHA 05]. Cependant les simulateurs actuels peinent à modéliser les conditions de tests sur route.

Dans cet article, nous relatons les expériences sur routes que nous avons menées afin d'évaluer les performances qu'il est possible d'attendre des communications inter-véhicules. Pour cela, nous avons mis en œuvre deux plates-formes de tests complémentaires, appelées Carman et Caremba. La première plate-forme est dédiée à la mesure de performances des couches réseaux dans les réseaux de véhicules. La deuxième plate-forme mesure les performances applicatives obtenues avec un protocole de routage dédié aux réseaux de véhicules [KHA 07b] et développé en espace utilisateur sur une architecture adaptée aux réseaux dynamiques [DUC 07a]. L'analyse des premiers résultats donne une idée de ce que les futures applications ITS pourront disposer en terme de débits ou de délais notamment.

La suite de l'article est structurée de la façon suivante. Dans un premier temps, nous résumons les travaux concernant les expériences sur route ainsi que les architectures embarquées sousjacentes aux applications ITS. Ensuite, nous détaillons l'architecture logicielle et matérielle de nos deux plates-formes expérimentales ainsi que les scénarios de tests avant d'analyser les premiers résultats obtenus. Nous terminons par quelques perspectives.

2. État de l'art sur les expérimentations

Différentes expérimentations ont concerné les VANETs. Elles utilisent principalement la norme IEEE 802.11. Dans [ABD 07], les auteurs démontrent en effet la faisabilité d'utiliser directement la norme IEEE 802.11b d'une voiture en mouvement connecté à un point d'accès dans le désert de Californie, où il n'y a aucun obstacle et aucune interférence radio et aucun autre véhicule. Dans [OTT 04], les auteurs ont étudié le comportement des connexions réseaux (TCP et UDP) pour une voiture en mouvement, connectée à un point d'accès. Le but était de comprendre l'impact de la vitesse de la voiture, du taux de transmission et de la taille des paquets. Dans ces expérimentations, des cartes IEEE 802.11b ont été utilisées et peu de précisions ont été révélées sur l'architecture embarquée utilisée.

Dans [BUC 05], les auteurs présentent quelques résultats expérimentaux utilisant une application multimédia dans un réseau ad hoc inter-véhicules. Ils ont utilisé deux véhicules équipés de la norme IEEE 802.11b et ont fait un ensemble de tests dans deux environnements différents (urbain et autoroute). Les auteurs arrivent aux conclusions suivantes : (i) le rapport Signal/Bruit (SNR pour *Signal to Noise Ratio*) est plus important sur autoroute que dans une zone urbaine, (ii) la liaison est plus fiable sur autoroute que dans une ville et (iii) il vaut mieux utiliser de grands paquets de données sur autoroute et de petits paquets (grande fragmentation) en zone urbaine. L'architecture utilisée repose sur Linux, des cartes PCMCIA, des GPS et le protocole UDP. Des résultats similaires sont obte-

nus dans [HUI 01] quant au SNR et au bruit. De plus, les auteurs analysent d'autres paramètres de performances tels que le RTT, le débit TCP et UDP, *etc.* Ils ont utilisé trois véhicules et un routage statique. Selon les auteurs, le déploiement d'applications multimédia est difficile dans un réseau de véhicules multi-saut. Leur architecture embarquée repose sur Windows XP, les protocoles UDP et TCP, iperf [IPE] pour la génération du trafic, un GPS, Netstumbler pour la mesure des signaux et un routage statique sur trois nœuds. Dans [SIN 02], les auteurs ont mesuré la qualité du lien sur autoroute dans des environnements urbains et semi-urbains. Les résultats de l'étude ont montré que l'environnement semi-urbain est le plus favorable pour la communication inter-véhicules. L'architecture utilisée par ces auteurs utilise Linux, un GPS, le protocole UDP et l'outil d'évaluation de performances réseau Netperf.

Le but des expériences faites par [GAS 06, OTT 04] est de comprendre les performances en termes de temps de connexion et de taux de perte quand une voiture mobile se connecte à des points d'accès. L'architecture embarquée utilisée dans [GAS 06] utilise Linux, les protocoles TCP, UDP et HTTP, iperf pour la génération du trafic TCP, wget et apache pour la génération de trafic HTTP et enfin tcpdump (couche réseau) et kismet (couche liaison) pour l'acquisition des données.

Ces divers expériences ne considèrent pas des scénarios de tests proches de la réalité (*i.e.*, croisement, dépassement ou multi-sauts entre les véhicules dans des scénarios hybrides tel que V2V et I2V). Un autre point important qui n'a pas été suffisamment traité est celui de l'influence de certains facteurs tel que la vitesse, l'accélération ou la distance sur les métriques de performance. Enfin, les expériences présentées dans cet article utilisent des routages statiques. Elles ne prennent pas en compte des protocoles de routage spécifiques aux réseaux de véhicules, ni des architectures embarquées adaptées aux applications devant évoluer en environnement dynamique. Or, les applications ITS reposeront très certainement sur des architectures logicielles spécifiques, intégrant des piles protocolaires adaptées. Divers projets de recherche portent sur les applications ITS aux USA (VII, CICAS, IVBSS...), en Europe (CVIS, SAFESPOT, COOPERS, PReVENT, GST, HIGHWAY, FLEETNET...), au Japon (SmartWay, VICS), en Inde (ITSIndia), en Allemagne (NOW), en France (PREDIT) *etc.* Actuellement, peu de projets ont spécifié leurs architectures embarquées ou publié des documents publics les concernant. L'architecture proposée par le projet GST (qui concerne les communications véhicules-infrastructure) se base sur Linux, OSGi [OSG], IPv6, HTTP, SOAP et le protocole OMA DM [GST]. L'architecture du projet CVIS [CVI] est basée sur Linux, OSGi, IPv6 et CALM. Plusieurs protocoles de routage spécifiques aux réseaux de véhicules sont à l'étude dans ces projets. Notons que les architectures devront rester ouvertes à de nouvelles solutions car le domaine est jeune, et la dynamique des réseaux de véhicules remet en cause de nombreuses solutions existantes, depuis les couches basses jusqu'aux applications en passant par la sécurité.

Cet article se propose de mesurer les performances des réseaux de véhicules dans divers scénarios réels sur route, avec et sans architecture et protocole spécifiques aux réseaux de véhicules.

3. Expérimentations avec la norme IEEE 802.11 sur route

Afin d'étudier les performances de la norme IEE 802.11 et de mieux appréhender les différents problèmes qui surviennent lors de la communication inter véhicules, nous avons procédé à de nombreux tests en condition réelle. Pour ce faire, nous avons mis en place notre propre plate-forme de test Caremba. Cette plate-forme est assez générique et permet de réaliser des tests sur route sur des réseaux de deux à six voitures, et de procurer les métriques permettant de valider ou non les scénarios de tests réalisés (*e.g.*, qualité du signal pour chaque trame émise sur chaque nœud, mesure du bruit, distance entre véhicules, vitesse et direction des véhicules, taux de perte, débit mesuré, quantité des données reçues, durée de connexion, gigue, temps d'aller retour).

3.1. *Description de la plate-forme Carman*

Lors des expérimentations, nous avons utilisé des voitures équipées du matériel présenté dans la figure 1. La plate-forme Carman (CAR-based Mobile Ad hoc Network) fonctionne sous Linux et a

Figure 1. *Matériel utilisé.*

été testée sur plusieurs distributions. En effet, suivant la distribution utilisée, certains outils réseaux sont nécessaires (*e.g.*, *ifconfig, ping,* les extensions et outils de *Jean Tourrilhes* [TOU], le package *Bluez* pour la connexion du GPS, *etc.*). Il faut noter également que la version du pilote *madwifi* [MAD] a été modifiée afin de permettre la capture d'informations avancées telles que la qualité du signal et le niveau de bruit. Plusieurs logiciels sont indispensables au fonctionnement de la plate-forme tels que le logiciel *Iperf* pour la génération de trafic réseau TCP ou UDP, VideoLan pour la visualisation et la diffusion des vidéos, GNU Plot [PLO] pour la création de courbes de résultats, *etc.* La figure 2 permet de visualiser l'architecture de Carman. Cette architecture met en exergue les

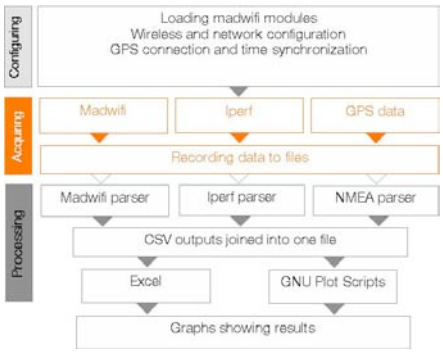


Figure 2. *Architecture de Carman.*

trois modules de cette plate-forme à savoir (i) un module pour la configuration de la plate-forme qui permet de charger les modules madwifi, de fixer les paramètres du réseau et de synchroniser les temps GPS/PC, (ii) un module pour l'acquisition des données qui est assez simple d'utilisation car le conducteur du véhicule peut aussi l'utiliser et enfin (iii) un module qui permet d'analyser les données capturées.

Au lancement du module d'acquisition des données, les différentes informations sont enregistrées dans des fichiers de logs. Au cours des expériences, les informations concernant les différentes métriques telles que le taux de perte et la qualité de signal sont affichées comme le montre la figure 3.

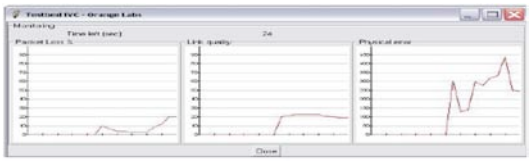


Figure 3. *Interface de moniteur.*

Dans le module d'analyse des données capturées, nous avons développé quatre analyseurs de trace (pour GPS, madwifi, Iperf et ping). Un script global a été crée pour appeler l'ensemble des analyseurs et fusionner les résultats. Ensuite le tracé des courbes est effectué grâce à GNU Plot.

3.2. *Scénarios des tests*

Les scénarios de tests envisagés ont pour but de couvrir le maximum des situations réelles. Nous allons évoquer les plus pertinents d'entre eux.

3.2.1. *Scénarios V2V*

Le premier scénario est le croisement de deux véhicules mobiles comme le montre la figure 4 (a) nous avons fait le test à faible vitesse 30 km/h et à vitesse modérée 50 km/h. Le scénario de

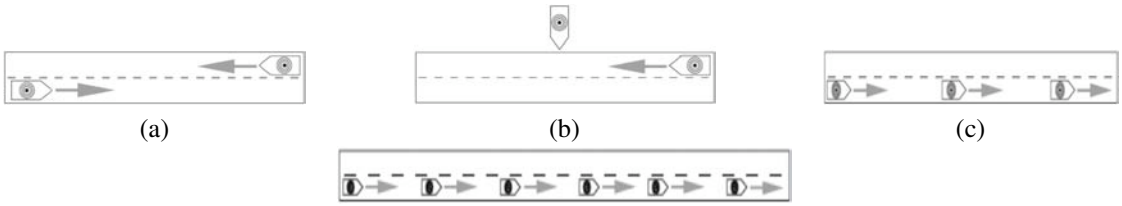


Figure 4. *(a) Croisement de deux véhicules, (b) Scénario de dépassement (b), (c) Scénario à trois véhicules, (d) Scénario à 6 véhicules.*

dépassement consiste au croisement de deux véhicules dont un est immobile (*cf.* figure 4 (b)). C'est une situation assez courante notamment dans les carrefours.

Le scénario d'accélération consiste à côtoyer deux véhicules puis faire accélérer l'un d'entre eux. Ceci permet de voir l'influence de l'accélération sur la communication.

Le scénario du simple saut est un scénario à trois véhicules qui permet de voir l'influence du passage par un nœud intermédiaire (*cf.* figure 4 (c)). Cette expérience a été faite avec des véhicules immobiles puis en mouvement avec deux vitesses prédéterminées (50 Km/h et 90 Km/h).

Le dernier scénario V2V testé est un scénario à six véhicules (*cf.* figure 4 (d)). Ce scénario permet d'étudier le comportement des communications utilisant quatre relais. Le choix des relais est préfixé grâce à l'affectation de routes statiques.

3.2.2. *Scénarios I2V*

Ces scénarios sont très importants et concernent le cas d'une infrastructure fixe proposant des services à automobilistes/passagers dans une ville ou sur une autoroute. Le scénario testé est basé sur une borne statique situé sur un pont alors qu'un véhicule circule sur une voie express passant en dessous. La borne émet des données de façon continue avec un débit maximal. Le véhicule, ne

connaissant pas le canal d'émission de la borne, est obligé de le découvrir via une recherche. Le but est de s'approcher au maximum des conditions réelles.

3.2.3. *Scénario hybride*

Ce scénario a pour objectif d'étendre la couverture d'une borne et représente un scénario hybride des deux précédents. Pour mettre en place ce scénario, la borne émettrice est située sur un pont alors que deux véhicules suffisamment proches pour pouvoir communiquer empruntent la voie express passant en dessous (cf. figure 5). Au début, la borne se connecte directement au premier véhicule et lui envoie les données. Lorsque celui-ci devient hors de sa portée, la borne tente de faire relayer les données par le second véhicule.

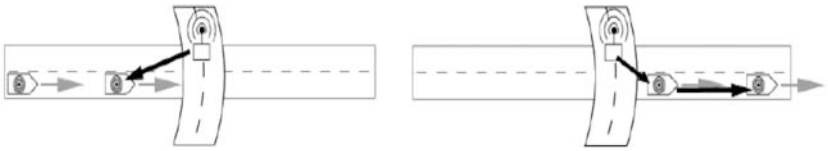
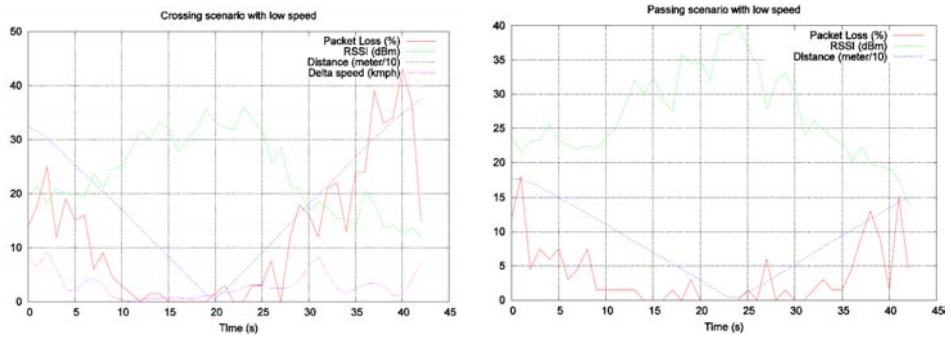


Figure 5. *Scénario hybride à deux véhicules.*

3.3. *Résultats obtenus et analyses*

3.3.1. *Expériences V2V*

Les résultats des scénarios V2V avec deux voitures sont illustrés par la figure 6 (dépassement et croisement). Grâce à ces scénarios, nous avons pu voir que la distance entre l'émetteur et le récepteur influe sur la qualité du signal et le taux de perte. Nous avons également pu constater que la vitesse affecte peu les performances des communications entre deux véhicules. La figure 7 montre un



(a) (b)

Figure 6. *Scénario de croisement (a) et de dépassement (b) à petite vitesse*

résumé des statistiques des premières expériences multi-sauts avec 3 voitures (statique et en suivi). En comparant les pertes moyennes obtenues et le délai d'aller-retour en petite ou grande vitesse, nous pouvons conclure que la vitesse affecte peu les performances des communications multi-sauts. On constate que les pertes dans le scénario de suivi sont inférieures au scénario statique malgré la mobilité de nœuds. Cette différence provient de l'environnement car sur autoroute, il n'y a ni obstacle ni interférence avec d'autres systèmes radios. La figure 8 résume les statistiques obtenues lors de la chaîne de communication de 6 véhicules, c'est-à-dire en utilisant 4 relais. Le taux de perte est de 15,69% de bout en bout soit environ 3,5% de perte par émission. En effet, les pertes

	Pertes moy (%)	RTT moy (ms)	Distance inter-vehiclule (m)
Scénario statique	4,75	4,41	87,17
Scénario de suivi (PV)	1,99	1,97	91,39
Scénario de suivi (GV)	3,48	2,04	142,05

Figure 7. *Résumé des statistiques des scenarios V2V*

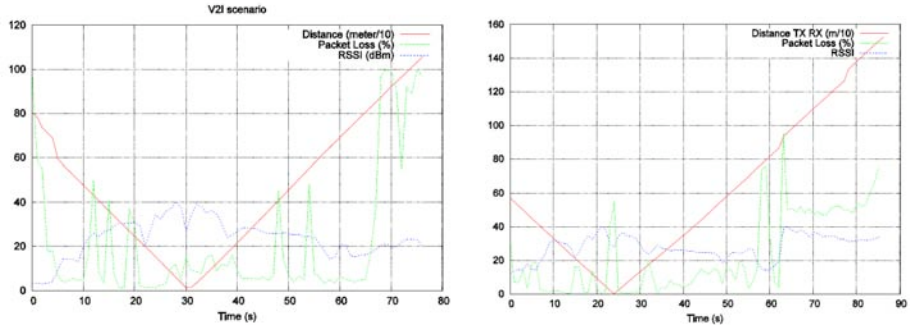
augmentent avec le nombre croissant de sauts radio. Ce phénomène s'explique par l'utilisation du mécanisme CSMA (plus le média est surchargé, plus la probabilité est grande que deux voitures émettent en même temps après avoir détecté un support libre augmentant ainsi le taux de pertes) et par le problème de station cachée (mécanisme RTS/CTS désactivé).

Distance inter véhicule m	71,93
Distance Total m	359,64
Vitesse km/h	78,69
Débit effectif kbps	252,67
Débit émetteur kbps	300
Taux de perte %	15,69
RTT ms	7,99
RSSI	42,71
Bruit dB	-98,79

Figure 8. *Moyenne des métriques de la chaîne de 6 véhicules*

3.3.2. *Expériences V2I*

Pour ces expérimentations, nous nous sommes focalisés sur les métriques suivantes : la puissance du signal, le taux de perte, la durée de connexion au réseau et le total de données reçues. Nous avons exécuté plusieurs fois les essais menant à des résultats semblables. La figure 9 représente la perte de paquet et le RSSI (mesure de la puissance reçue d'un signal radio) pour le scénario simple avec un véhicule et un point d'accès (a) et pour le scénario hybride avec deux véhicules et un point d'accès (b). L'impact des ponts est clairement visible. La figure 10 montre le temps de connexion et



(a) (b)

Figure 9. *Taux de perte et RSSI pour le scenario simple (a) et hybride (b) (grande vitesse).*

la quantité de données transférées pour les deux derniers scénarios simple et hybride. Les résultats expérimentaux ont révélé la faisabilité du réseau ad hoc pour étendre la zone de couverture des points

	Temps de connexion		Quantité de données	
	Grande vitesse	Petite vitesse	Grande vitesse	Petite vitesse
Total	81	140	59.195	98.86

(a)

	Temps de connexion		Quantité de données	
	Grande vitesse	Petite vitesse	Grande vitesse	Petite vitesse
Direct	66	143	54.247	106.452
Multihop	25	27	9.813	9.519
Total	91	170	64	116

(b)

Figure 10. *Durée de connexion (en secondes) et quantité de données (en méga-octets) transférées dans le scénario simple (a) et hybride (b)*

d'accès. Le réseau ad hoc accroît ainsi la durée de connexion pour des voitures en mouvement de plus de 16%. Il augmente également la quantité données que la voiture peut recevoir de plus de 10%.

4. Expérimentations avec un protocole de routage et une architecture dédiée aux VANET

Dans cette section, nous complétons les expérimentations précédentes en présentant des tests sur route réalisés avec une architecture logicielle embarquée et un protocole de routage dédié aux réseaux de véhicules. Les mesures sont réalisées au niveau applicatif, grâce à une application qui génère du trafic et qui mesure précisément les délais de transmission.

4.1. *Description de la plate-forme Caremba*

La plate-forme Caremba est constituée d'un certain nombre de kits composés de PC (PC *shoebox* industriels 12V ou portables), d'un récepteur GPS, d'une carte IEEE 802.11 et d'une antenne WiFi externe pour les communications (*cf.* figure 11). Notons que les antennes utilisées pour les tests ne sont pas toutes identiques (*i.e.,* puissance d'émission différentes).

Figure 11. *Plate-forme de test : Caremba.*

Les PC utilisent Linux et l'architecture logicielle Airplug. La suite logicielle Airplug [DUC 07a] a été développée de manière à permettre les interactions intra ou inter véhicules entre applications embarquées tout en garantissant une grande robustesse face aux défaillances. Elle est constituée d'un programme cœur (`airplug`) qui gère les communications et sur lequel sont connectés divers processus (*cf.* figure12-a). Les communications locales (communications entre différents processus sur la même machine) sont réalisées par des tubes. Les communications distantes font appel à des sockets utilisant divers protocoles de communication (des couches 2, 3 et 4, ou des couches 2 uniquement). De cette manière, chaque processus possède son propre espace mémoire, et seul `airplug` accède au réseau, ce qui permet diverses optimisations (*pyggybacking*, ordonnancement...) et renforce la robustesse de part l'indépendance des processus développés en mode utilisateur. En outre, cette architecture n'impose aucun paradigme ni langage de programmation puisque les applications doivent

simplement savoir lire leur entrée standard et écrire sur leur sortie standard. Airplug propose un routage inter-applications adapté aux réseaux dynamiques. Les applications peuvent s'abonner aux flux d'information produites par leur consœurs, ou bien envoyer une requête et attendre une réponse. Les messages peuvent être envoyés localement et/ou dans le voisinage, à une application particulière ou à toutes les applications abonnées.

Par ailleurs, cette architecture permet le développement de protocoles en mode utilisateur, ainsi que la cohabitation de plusieurs protocoles (*e.g.*, routage géographique pour les alertes, routage dirigé par la source pour la connexion à un point d'accès infrastructure). Les expériences reportées dans cet article ont utilisé les *transmissions conditionnelles* [KHA 07b], implémentées sous la forme d'une application compatible Airplug appelée HOP (*cf.* figure 12-b). Pour ces expériences, les communications inter-véhicules ont utilisé des socket UDP. Les transmissions conditionnelles remplacent l'adresse de destination par des conditions telles que la distance à l'émetteur, la trajectoire, la zone géographique *etc.* Grâce à l'évaluation dynamique de ces conditions, les performances obtenues sont très bonnes comparativement aux adresses réseaux ou géographiques, dont la pérénité est faible dans un réseau dynamique.

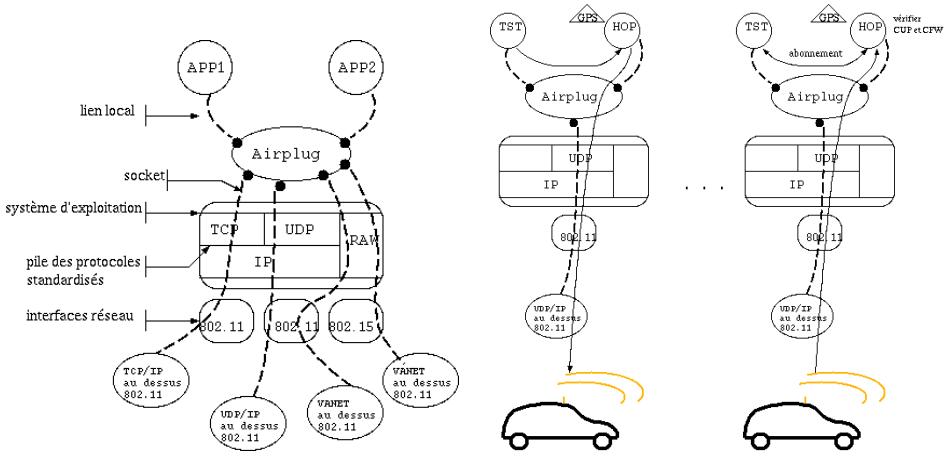


Figure 12. *(a) Architecture embarquée Airplug, (b) Utilisation de HOP pour les tests.*

4.2. *Fonctionnement et scénarios*

Pour le déroulement de ces expériences, une seule application TST (embarquée dans la voiture de tête) envoie des données (*cf.* figure 12 (b)). Cette application envoie localement à l'attention de HOP (via airplug) le message à transmettre et les conditions de transmission `CUP` et `CFW`. Les champs de contrôle `CUP` et `CFW` correspondent respectivement à la condition de réception du message, et à sa retransmission par l'application réceptrice. Une fois que l'application HOP reçoit ce message (via son entrée standard), elle envoie à `airplug` le message à diffuser. Le programme `airplug` diffuse alors ce message vers les autres applications HOP embarquées dans les autres voitures. Le message transmis contient les différents champs de contrôle dont les conditions `CUP` et `CFW`.

Dans les autres voitures, chaque application TST s'abonne à l'application HOP locale en lui envoyant (via airplug) un message de contrôle spécifique. Dès que le programme `airplug` reçoit un message, il vérifie l'application destinatrice (ici HOP) et l'envoie en local. Une fois que l'application HOP récupère le message qui lui a été envoyé sur le réseau, elle vérifie dynamiquement les conditions

de transmission. Selon le résultat de ces évaluations, il transmet localement le message reçu aux applications abonnées à ses messages (ici TST) et/ou retransmet à nouveau ce message au réseau.

Dans ces expériences, les cinq véhicules sont en convoi (figure 13). Le programme TST de la voiture leader envoie à intervalle régulier des paquets de taille fixe. Il combine les datations GPS (reçues à la fréquence d'un hertz) avec les horloges locales des ordinateurs (non synchronisées) de manière à permettre une mesure fine des délais de transmission. Nous avons choisi des conditions pour `CUP` et `CFW` de telle sorte qu'on fixe une seule direction de transmission des messages (*i.e.,* voiture de tête vers voiture de queue). Si la voiture réceptrice est derrière l'émettrice alors HOP (i) transmet localement le message reçu à l'application TST abonnée à ses messages et (ii) retransmet à nouveau ce message dans le voisinage. Chaque applications HOP reçoit la position GPS *via* une application locale qui lit les trames NMEA envoyées par un récepteur GPS.

Les conditions `CUP` et `CFW` permettent de fixer une direction de transmission mais nous avons constaté que les messages ne faisaient pas tous autant de rebonds à cause des antennes hétérogènes et de l'environnement changeant. Par exemple, la troisième voiture recevait souvent des messages directement de la voiture leader (figure 13 haut). Dans une seconde expérience, nous avons contraint les messages à passer par toutes les voitures successives grâce aux conditions, même si la portée aurait permis d'éviter un saut (figure 13 bas).

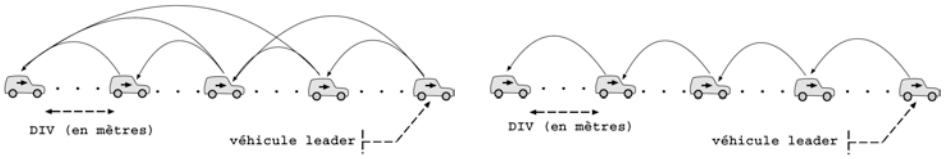


Figure 13. *Scénarios.*

4.3. *Résultats et analyses*

Dans le premier test, les résultats obtenus sont en terme de distance inter-véhicules (DIV). L'hétérogénéité des antennes explique en partie les résultats obtenus dans le tableau de la figure 14. En effet, la voiture 3 reçoit directement les paquets de la voiture 1 d'où un délai et un taux de pertes équivalent à la voiture 2. De plus, vu que la voiture 4 a une antenne de portée de communication inférieure à celle de la voiture 5, elle reçoit les paquets après la voiture 5. En effet, la voiture 5 reçoit les paquets aussi bien de la voiture 4 que de la 3 ou de la 2 et elle peut être considérée à 2 ou 3 sauts de la voiture 1 (d'où des délais largement inférieur à ceux enregistrés dans la voiture 4). Notons aussi que la portée des antennes a été très variable en fonction du trafic et de l'environnement. En conclusion, il a été difficile d'obtenir des résultats en fonction de la distance.

Dans la deuxième expérience, en forçant les rebonds, nous avons obtenu des résultats en fonction du nombre de sauts. Le nombre de sauts impacte directement les performances de la réception (figure 15). Par exemple, le délai moyen de réception augmente linéairement (coefficient directeur de $+246$ ms).

5. Conclusion et Perspective

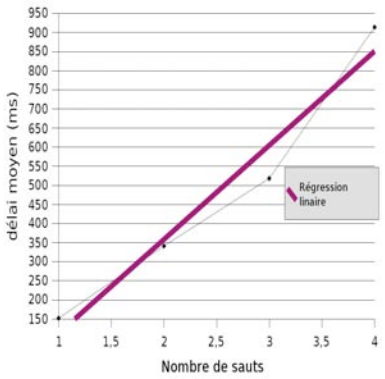
Dans cet article, nous nous sommes intéressés aux performances réelles des réseaux de véhicules. En utilisant deux plates-formes, nous avons obtenu des performances brutes avec le WiFi et un routage statique, et des performances applicatives avec un routage VANET.

PARAMETRES				
taille paquets ko	500			
total données émises ko	900.1			
voitures	2	3	4	5
DIV moyen m	57	103	182	268
pertes %	13	15	50	12
intervalles de pertes	279	311	667	274
délai 1^{er} paquet	971	979	1814	665
délai moyen ms	88	68	2722	884
total données reçues ko	778	767	453	788

Figure 14. *Résultats obtenus selon la distance inter-véhicules DIV.*

PARAMETRES				
taille paquets ko	500			
total données émises ko	1000			
nombre de sauts	1	2	3	4
pertes %	9	32	41	58
intervalles de pertes	97	207	184	364
délai 1^{er} paquet	95	132	260	386
délai moyen ms	154	341	518	914
total données reçues ko	694	636	652	578

(a)



(b)

Figure 15. *(a) Résultats obtenus selon le nombre de sauts, (b) Régression linaire (délai moyen)*

La première plate-forme Carman nous a permis d'expérimenter au niveau des couches réseaux en utilisant un routage statique. Cette plate-forme reste assez facile à utiliser et permet de procurer plusieurs métriques nécessaires pour une caractérisation fine des communications dans les réseaux véhiculaires (*i.e.*, qualité du signal, mesure du bruit, taux de perte, débit, durée de connexion, gigue, temps d'aller retour *etc.*). La deuxième plate-forme nous a permis d'expérimenter un protocole de routage dédié aux VANET (les transmissions conditionnelles) dans un environnement applicatif adapté aux réseaux dynamiques (Airlug).

Ces deux plates-formes nous ont permis d'effectuer des tests mettant en place des réseaux allant de deux à six véhicules. Les premiers résultats obtenus lors de ces tests sont encourageants. En particulier, ils montrent la bonne performance de la norme 802.11 avec des pertes ou des délais relativement faible. Nous avons également constaté que la distance est un facteur de perte alors que la vitesse et l'accélération ne sont que peu influentes sur la communication. Les résultats expérimentaux ont révélé aussi la faisabilité du réseau ad hoc pour étendre la zone de couverture des points d'accès. Les premiers résultats obtenus au niveau applicatif sont encourageants. Le délai moyen après 5 sauts reste inférieur à la seconde. Par contre, le délai inter-paquets devra être ajusté afin de limiter les collisions dans le convoi. Ces travaux sont en cours.

Par la suite, nous prévoyons de compléter ces premiers résultats avec d'autres mesures et d'autres scénarios (communication de véhicules avec plusieurs points d'accès, évaluation des performances du handover *etc.*), d'évaluer les performances d'autre protocoles de routage et d'auto-organisation conçus pour les réseaux de véhicules (*e.g.*, GyTAR [JER 07]) et de tester des applications ITS sur route (détection de convoi, prévisibilité de route, suivi de véhicules, *chat...*).

Remerciements

Les auteurs tiennent à remercier Yacine Khaled et Patrick Marlier pour leurs contributions dans la conception et le développement des plates-formes de test. Merci également aux personnes de France Télécom et de l'UTC qui nous ont aidés pour ces tests.

6. Bibliographie

[ABD 07] ABDALLA G., ABU-RGHEFF M., SENOUCI S., « Current Trends in Vehicular Ad Hoc Networks », in UBIROADS 2007 workshop, IEEE GIIS Marrakech, Morocco, July 2007.

[BUC 05] BUCCIOL P., MASALA E., KAWAGUCHI N., TAKEDA K., DE MARTIN J., « Performance Evaluation of H.264 Video Streaming over Inter-Vehicular 802.11 Ad Hoc Networks », Proc. of 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), Berlin, Germany, Sep. 2005.

[CVI] « CVIS making the connection. », http ://cvis.odeum.com/download/cvis_first_brochure.pdf.

[DUC 07a] DUCOURTHIAL B., « About efficiency in wireless communication frameworks on vehicular networks », Invited paper, workshop ACM WIN-ITS co-located with IEEE ACM QShine 2007, Canada, August 2007.

[DUC 07b] DUCOURTHIAL B., KHALED Y., SHAWKY M., « Conditional transmissions, a strategy for highly dynamic vehicular ad hoc network. », In IEEE WoWMoM'07, 2007.

[GAS 06] GASS R., SCOTT J., DIOT C., « Measurements of In-Motion 802.11 Networking », In Proc. WMCSA, Apr. 2006.

[GST] « GST open systems architecture and interface specifiaction. », http ://www.gstproject.org/os/documents/DEL_GST_OS_DEV_Reference_Implementation_3_3_2.pdf.

[HUI 01] HUI F., « Experimental Charaterization of Communications in Vehicular Ad Hoc Networks », Thesis Master of Science in Computer Science, University of California Davis, 2001.

[IPE] « IPERF Traffic Generator. », http ://dast.nlanr.net/Projects/Iperf/.

[JER 07] JERBI M., SENOUCI S., MERAIHI R., Y. G.-D., « An improved Vehicular Ad Hoc Routing Protocol for City Environments », IEEE International Conference on Communications (ICC), Glasgow, Scotland, June 2007.

[KHA 05] KHALED Y., DUCOURTHIAL B., SHAWKY M., « IEEE 802.11 performances for inter-vehicle communication networks », *Vehicular Technology Conference,2005. VTC 2005-Spring*, vol. 5, 2005, p. 2925-2929.

[KHA 07a] KHALED Y., DUCOURTHIAL B., SHAWKY M., « A usage oriented taxonomy of routing protocols in VANET », June 2007.

[KHA 07b] KHALED Y., DUCOURTHIAL B., SHAWKY M., « Conditional transmissions : a communication strategy for highly dynamic vehicular ad hoc networks », *IEEE Transactions on Vehicular Technology, special issue on vehicular communication networks,*, vol. 56, n° 6, November 2007, p. 3348-3357.

[MAD] « Multi-band atheros driver for wifi (madwifi) », http ://madwifi.org/.

[OSG] « Wire admin service, OSGi package », org.osgi.service.wireadmin.

[OTT 04] OTT J., KUTSCHER D., « Drive-thru internet : IEEE 802.11b for automobile users », In IEEE INFOCOM, 2004.

[PLO] « GNU Plot », http ://www.gnuplot.info/.

[SIN 02] SINGH J., BAMBOS N., SRINIVASAN B., D. C., « Wireless lan performance under varied stress conditions in vehicular traffic scenarios. », *In Proceedings. IEEE 56th Vehicular Technology Conference*, vol. 2, 2002, p. 743-747.

[TOU] « Wireless for linux. », http ://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/.

Gestion Autonomique de la Répartition de Charge pour le Contrôle de la Congestion dans les Réseaux Sans Fil

Gilbert Sawma[*], Ryad Ben-El-Kezadri[* **], Issam Aib[***], Guy Pujolle[*]

[*] *Laboratoire d'Informatique de Paris 6, Paris Universitas*
104 Avenue du President Kennedy, 75016, Paris, France
[**] *Laboratoire CRISTAL/ENSI, Université de la Manouba*
2010, Manouba, Tunisie
{Gilbert.Sawma, Ryad.Bek, Guy.Pujolle}@lip6.fr
[***] *D.R. Cheriton School of Computer Science, University of Waterloo*
200 University Avenue West, Waterloo, Ontario
iaib@uwaterloo.ca

RÉSUMÉ. *Dans cet article, nous proposons un système de gestion autonomique des réseaux 802.11 à l'aide d'un algorithme de répartition de charge baptisé ALBA. L'idée fondamentale et originale d'ALBA est sa capacité à réorganiser localement le réseau et répartir dynamiquement et au mieux sur une zone les stations sur les points d'accès, afin d'assurer l'efficacité du service et la satisfaction des clients. L'autonomicité intervient dans le système afin d'enrichir la connaissance du réseau et de fournir à chaque point d'accès une vue située de son voisinage et du trafic des clients. L'évaluation de la solution révèle des gains importants de performance en terme de capacité d'accueil en stations du réseau.*

ABSTRACT. *In this article, we propose an autonomic management system for 802.11 networks through a load-balancing algorithm called ALBA. The original idea of ALBA is its ability to reorganize the network locally and to dynamically allocate the stations to the access points, in order to ensure the efficiency of services and customer satisfaction. The autonomic concept is involved in the system in order to enrich the knowledge of the network and to provide each access point a situated view of its neighborhood and clients' traffic. The evaluation of the solution reveals significant gains in terms of capacity in the network.*

MOTS-CLÉS : *IEEE 802.11, VoIP, Autonomique, Contrôle d'Admission, Répartition de la charge.*

KEYWORDS: *IEEE 802.11, VoIP, Autonomic, Admission Control, Load balancing.*

1. Introduction

Cet article présente un nouvel algorithme de distribution de charge (Load Balancing) pour les grands réseaux WLAN (Wireless Local Area Networks WLAN). Ces réseaux d'infrastructure, souvent très denses, sont ceux couvrant, les campus universitaires, les réseaux d'entreprise et de façon générale tous les hots spots. A l'heure où les échanges multimédia se généralisent et la demande en débit s'accroit, le support de la voix sur IP (VoIP) sur WLAN (VoWLAN) constitue un indicateur de choix sur la capacité d'un réseau 802.11 à satisfaire à la demande de ses usagers. Le nombre d'appels VoIP supporté par une cellule 802.11b est aujourd'hui limité entre 3 à 12 appels bidirectionnels simultanés [GAR 03b] [GAR 03a]. Le problème des WLAN est que la capacité globale du réseau est réduite du fait de la fragmentation du réseau en multiples cellules [IEE 97]. Sans politique efficace d'allocation des Stations (STA) aux Points d'Accès (AP), peuvent se former au milieu de zones réseau sous peuplées, des îlots d'APs surchargés de stations, conduisant ainsi à une saturation locale du réseau. Plusieurs études montrent à ce titre que la congestion dans les réseaux d'infrastructure 802.11 est un phénomène localisé sur quelques APs [BAL 03] [BAL 02b]. Contrairement aux algorithmes existants qui fonctionnent localement [BAL 02a] [CIS 04] [SHE 99] [VEL 04], la solution que nous proposons contrôle les connexions de tous les utilisateurs sur un voisinage étendu. La répartition intelligente de la charge aboutit à une augmentation certaine de l'utilisation du réseau et à la satisfaction d'un plus grand nombre d'utilisateurs. Pour ce faire, l'algorithme redistribue dynamiquement les stations sur les APs localement présents pour résoudre les congestions spontanées.

Les possibilités de réassociation étant limitées au voisinage proche des APs, nous exploitons dans notre solution les caractéristiques des systèmes autonomiques et plus particulièrement la notion de *Plan de Connaissance*. Le *Plan de Connaissance* des réseaux autonomiques vient s'ajouter aux plans de données, de contrôle et de management des architectures traditionnelles. Ce nouveau plan est conçu pour collecter et stocker des informations importantes. Il est alimenté par des composants de mesures incorporés dans les différentes machines. Le *Plan de Connaissance* fournit une vue située. Par définition, la vue située d'un noeud constitue l'ensemble des informations ou connaissances collectées, pour un ensemble d'algorithmes, des noeuds voisins positionnés à un nombre de sauts donné. Ces informations sont actualisées et mises à jour selon la fréquence souhaitée par les algorithmes.

Notre solution baptisée ALBA (Autonomic Load Balancing Algorithm) consiste à déplacer les clients des APs surchargés vers leurs voisins non saturés afin de permettre à de nouvelles stations de s'insérer dans le réseau. La réassociation à basse vitesse d'une station pouvant pénaliser les stations déjà présentes dans une cellule [HEU 03], la faisabilité de chaque réassociation doit être évaluée. Dans ce but, ALBA maintient une vue située de l'environnement d'AP afin de garantir que la rétrogradation en débit d'une station n'induit pas de dégradation de QoS pour les autres clients.

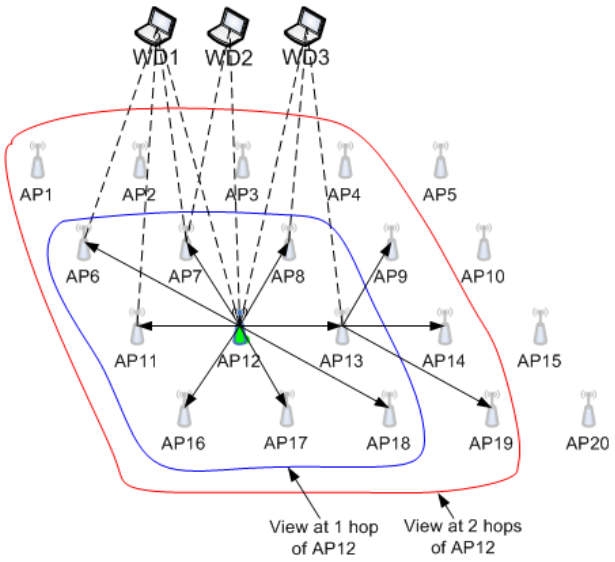


Figure 1. *Topologie du système de distribution. Les flèches montrent la décharge centrifuge de AP12 et AP13 respectivement. WD1, WD2 et WD3 sont visibles par plus d'un AP comme l'indiquent les traits en pointillés.*

La métrique principale entretenue par le plan de connaissance est l'estimation de l'utilisation du canal *CUE* (Channel Utilisation Estimate) [GAR 03c]. Cette métrique fournit aux APs une estimation précise de la charge des cellules environnantes.

Ce papier est structuré comme suit. La section 2 présente un état de l'art des différents mécanismes existants pour la distribution de charge. Dans la section 3, nous présentons les aspects de la vue située relatifs au plan de connaissance d'ALBA. La section 4, présente l'algorithme de la distribution de la charge et son utilisation du plan de connaissance. La section 5 présente les résultats de simulation. La section 6 offre un résumé des travaux et présente nos perspectives de recherches.

2. Travaux reliés à la distribution de la charge

La charge utilisateur étant souvent répartie de façon inégale entre les cellules dans les WLANs, le thème de la répartition de la charge fait l'objet de plusieurs études [CIS 04] [SHE 99] [VEL 04] [DAH 06] [BEJ 04]. Les algorithmes de répartition de charge se basant essentiellement sur la métrique d'indication de charge pour réaliser des projections et déclencher la redistribution des stations, le choix de l'estimateur de charge constitue un élément fondamental dans leur conception. La métrique de charge détermine également le seuil limite de congestion représentant l'alarme de déclenchement de l'algorithme d'équilibrage de charge.

Dans [CIS 04], les auteurs prennent en considération la bande passante courante des APs et associent les utilisateurs aux APs les moins chargés. Un algorithme prédictif d'association fondé sur la puissance du signal reçu RSSI (Received Signal Strength Indicator) est étudié dans [SHE 99]. Selon cet algorithme, les APs renvoient la valeur RSSI moyenne des stations qui leur sont connectées et les stations utilisent cette valeur pour choisir l'AP pour lequel la dégradation de RSSI est la moins grande. L'inconvénient de [CIS 04] et [SHE 99] est de prendre en compte un seul paramètre, et non pas l'ensemble en même temps, i.e., la puissance du signal reçu des APs et leurs charges ne sont pas considérés simultanément.

[VEL 04] utilise le débit comme métrique pour déterminer la charge des APs. Le problème de la métrique de débit pour déterminer la charge restante est de ne pas prendre en compte les conditions particulières de fonctionnement du réseau telles que le débit physique des transmissions et le type des applications utilisées. [GAR 03b] [GAR 03a] démontre à ce titre que le débit ne peut pas être utilisé pour définir la capacité du réseau. Les expérimentations et les calculs montrent en effet que la charge maximale utile d'une cellule est d'environ 6.2 Mbps pour le trafic de données, mais seulement de 800 Kbps pour un réseau VoWLAN utilisant le codec G711 avec 10 ms de données audio par paquet. [GAR 03c] propose de fait l'utilisation de la métrique d'estimation d'utilisation du canal CUE à la place du débit.

[DAH 06] propose un algorithme d'équilibrage à portée locale, utilisant une architecture centralisée. Les expérimentations sont réalisées sur un réseau de 4 APs. Les problèmes de ce travail sur le plan de la validation sont de restreindre l'évaluation à de petits réseaux et sur le plan conceptuel de se baser sur une architecture centralisée limitant son passage à l'échelle.

Au niveau des organismes de normalisation, le groupe de travail dédié 802.11v a été créé pour proposer un amendement à la norme 802.11 relatif à la gestion de réseaux sans fil (Wireless Network Management WNM). Les discussions du groupe de travail 802.11v sont toutefois limitées à la normalisation des exigences des algorithmes, et non aux algorithmes d'équilibrage de charge eux-mêmes. L'approbation définitive de cet amendement n'est pas prévue avant 2008.

Notre solution se distingue des travaux précédemment cités par i) l'utilisation de la métrique de charge CUE pour garantir la QoS des flots, ii) l'approche autonomique basée sur le Plan de Connaissance, l'architecture distribuée et la vue située, iii) le traitement atomique des opérations de redistribution des stations dans la vue située.

3. Le Plan de Connaissance dans les WLANs

La gestion autonomique des réseaux est un domaine d'étude actif faisant l'objet de nombreuses recherches. La gestion des réseaux sans fil est d'autant plus importante que ces systèmes supposent un nombre important d'équipements, doivent être simple à installer et être capable de s'adapter à des conditions environnementales variables. Un système autonome est généralement défini en termes de fonctions auto-

nomes qualifiées de " self-* " comme par exemple les fonctions d'auto management (self-management) qui limitent le nombre d'intervention humaine dans la configuration. Au niveau architecturel, l'innovation fondamentale des systèmes autonomiques est l'introduction d'un quatrième plan réseau venant s'ajouter aux trois plans traditionnels (plan de donnés, plan de contrôle, plan de gestion). Selon les études, ce plan est appelé plan d'information, plan de gouvernance ou plan de connaissance.

Dans le domaine des réseaux sans fil, le standard IEEE 802.21, en cours d'élaboration, utilise ce $4^{ème}$ plan, appelé "Plan d'Information", pour collecter les différentes informations des réseaux sans fils disponibles et permettre les décisions de transfert (handover) entre les différentes technologies du marché. Dans les réseaux coopératifs, le "Plan de Coopération" assure le service de coopération entre les différents éléments du réseau. Dans les réseaux autonomiques ce plan est parfois appelé "Plan de Gouvernance ou Inférence" [STR 06].

Le concept de *Plan de Connaissance* dans les réseaux a été introduit en 2003 [CLA 03]. [CLA 03] suggère un nouveau concept réseau permettant à un réseau de se configurer lui-même en se basant sur des instructions de haut niveau, de détecter les anomalies, de les réparer, et de rendre un rapport de mise à jour à chaque fois que des changements interviennent dans les besoins. Pour atteindre cet objectif, les auteurs font valoir qu'il ne suffit pas d'améliorer progressivement les techniques et algorithmes connus. Au lieu de cela, ils proposent une nouvelle architecture avec un Plan de Connaissance (Knowledge Plane), omniprésent dans le réseau, capable de fournir l'état de l'environnement opérationnel des différents éléments du réseau. Le plan de connaissance que nous proposons dans ALBA se base sur la "Vue Située", voir figure 1. La vue située d'un noeud *x* est l'ensemble des informations ou connaissances rassemblées, par chaque algorithme, des noeuds voisins situés à *n* sauts au plus. Le plan de connaissance sonde l'état du réseau d'accès, c'est-à-dire, l'état actuel des ressources, les points d'accès sans fil, les stations associées, leurs vitesses physiques, les caractéristiques du trafic de chaque application, etc. Cette connaissance est utilisée par ALBA lors d'une redistribution de charge pour choisir vers quels APs réassocier quelles stations.

L'ensemble des informations disponibles dans le plan de connaissance de chaque AP pour réaliser les prévisions de CUE lors d'une redistribution de charge dans le voisinage, sont les suivantes :

– Chaque AP_i maintient la liste de ses voisins. Les voisins sont classés selon leur distance à AP_i. La liste $AP.neighborList(n)$ désigne les APs placés à *n* sauts de AP_i. Cette connaissance peut être obtenue à l'aide d'un protocole de découverte des voisins dans le Système de Distribution du réseau. Nous avons ainsi, en se référant à la figure 1.

$AP_{12}.neighborList(3)$={AP_1,...,AP_{20}}, $AP_{16}.neighborList(1)$={AP_{11}, AP_{12}, AP_{17}}.

– Chaque AP_i maintient la liste des STA_s visibles, c'est-à-dire l'ensemble des stations générant une activité sur le canal. Ces informations sont obtenues en analysant

le trafic sur le canal. La liste de ces stations est notée $AP.visibleSTA_s$. Par exemple, en référence à la figure 1,

$AP_{12}.visibleSTAs$={STA_1, STA_2, STA_3}.

– Chaque AP_i maintient la liste des STA_s qui lui sont associées. Ces informations sont extraites du système des APs. La liste de ces stations est un sous ensemble de $AP.visibleSTA_s$ et est notée $AP.associatedSTA_s$. En référence à la figure 1,

$AP_{12}.associatedSTA_s$={STA_1, STA_3}.

– Chaque AP_i maintient enfin pour chaque STA le débit physique et les applications utilisées. Les informations de débit sont extraites du système des APs et le type des applications de l'analyse du trafic sur le canal.

Le tableau 1 montre à titre d'exemple la connaissance maintenue dans l'AP12.

Tableau 1. *Exemple de la vue située de l'AP_{12} dans le cas classique / avec ALBA*

visibleSTAs	**Application**	**Débit physique**	**CUE estimé**	**associatedSTAs classique**	**associatedSTAs ALBA**
STA_1	VoIP	11	0.0815	Oui	Non
STA_2	VoIP	5.5	0.1	Non	Oui
STA_3	VoIP	1	0.268	Oui	Oui

neighborList (n)
n=1 saut ; APs : 6, 7, 8, 11, 13, 16, 17, 18
n=2 sauts ; APs : 1...4, 6...9, 11, 13, 14, 16...19
n=3 sauts ; APs : 1...20

4. Algorithme de distribution de charge ALBA (Autonomic Load Balancing Algorithm)

Dans les WLAN, la portée des points d'accès dépend de la robustesse du codage choisi. Un codage robuste permet la couverture d'une zone plus grande mais avec un débit plus faible. Afin d'offrir le meilleur service à leurs clients, les opérateurs de hotspots réalisent des déploiements denses dans lesquels chaque parcelle du territoire est couverte à haut débit à l'aide d'au moins un AP. En théorie des graphes, ce type de déploiement correspond à une 1-couverture du territoire à haut débit. Dans les réseaux d'infrastructure IEEE 802.11, sous des hypothèses de fonctionnement ordinaires, une 1-couverture à 11Mbit/s de l'espace induit au minimum une 9-couverture à 1Mbits/s. Un réseau ordinaire offre donc la possibilité à chaque client de choisir à chaque endroit de l'espace et à chaque instant un AP parmi au minimum 9 disponibles. Il existe donc potentiellement 9^N combinaisons de connecter N clients dans le coeur d'un réseau IEEE 802.11 en mode infrastructure. L'objectif d'ALBA est de désengorger un voisinage d'AP dans une vue située en choisissant une combinaison de connexions garantissant les exigences de QoS des clients exprimées en terme de CUE. Une valeur de CUE est associée à chaque flot applicatif. Celle-ci peut être obtenue à partir de mesures de trafic ou déduite directement du type de l'application. La dernière méthode

est utilisée par défaut. L'avantage de cette méthode est de rendre des valeurs à partir desquelles des prédictions fiables peuvent être réalisées. Dans ce papier, nous testons la capacité de notre algorithme à traiter des flux VoIP à exigences connues. L'algorithme ALBA réagit de façon réactive et se déclenche lorsque le seuil d'utilisation Th_{alba} de l'AP est atteint. Dans le cas où l'algorithme ne parvient pas à trouver une combinaison de connexions satisfaisantes, la demande d'appel est rejetée.

4.1. *Métrique d'utilisation du canal CUE [GAR 03c]*

[GAR 03a] [GAR 03c] montrent que la mesure de la capacité restante du réseau en terme de bande passante n'est pas suffisante pour déterminer si un flot de charge donnée peut s'insérer sans provoquer de congestion. La possibilité d'insérer un flot dépend également du nombre d'applications présentes, de leur type et de l'overhead. La métrique CUE prenant en compte ses paramètres, nous l'utilisons dans la suite.

Nous illustrons par la suite les facteurs principaux dont le calcul de CUE dépend directement.

CUEtotal désigne la somme de tous les CUE des flux d'une cellule. Le CUEtotal s'écrit [GAR 03c] :

$$CUEtotal = \sum_{i \in flots} CUE(i) \quad [1]$$

Un trafic VoIP est composé d'un flux montant (upstream) et d'un flux descendant (downstream). Nous obtenons donc :

$$CUE(i) = 2 \times \frac{T_b}{T} \quad [2]$$

Où T est l'intervalle de temps périodique sur lequel se base la mesure du CUE (supposé 1 seconde dans notre travail) et T_b est le temps nécessaire à la transmission des données VoIP. T_b est fonction du temps de transmission et des overheads. T_b s'écrit :

$$T_b = n \times (T_P + T_O) \quad [3]$$

Où n est le nombre des trames transmises pendant le temps T (ici T=1 sec, donc n=50 trames avec le codec G711 à 20ms), T_P le temps total de transmission du payload et T_O celui des overheads dûs aux différentes entêtes, aux backoff et aux intertrames 802.11 ainsi qu'aux collisions.

Le tableau 2 montre les différentes valeurs de CUE induites par un flot VoIP pour les différentes vitesses physiques de 802.11b.

La capacité d'un AP vaut 1 (100%) et représente la valeur maximale de CUEtotal. Les collisions consommant au pire des cas 15% du temps d'occupation du canal [GAR 03c], nous fixons à 85% le seuil de déclenchement Th_{alba} de l'algorithme.

Tableau 2. *Exemple de CUE à 11, 5.5, 2 et 1 Mbps*

Vitesse (bps)	Codec (bps)	Payload (ms)	Payload (Bits)	$T_P(\mu s)$	$T_O(\mu s)$	$T_b(ms)$	CUE
11*10^6	64000	20	1280	116	698	40	0.0815
5.5*10^6	64000	20	1280	233	768	50	0.1
2*10^6	64000	20	1280	640	1012	82	0.165
1*10^6	64000	20	1280	128	1396	13	0.268

Sous ALBA, un flot VoIP est admis directement, si la valeur CUEtotal des flots déjà présents dans la cellule plus celle du nouveau flot est inférieure à 85%. Dans le cas contraire ALBA redistribue les connexions sur le voisinage de l'AP selon une heuristique donnée. Cette redistribution est réalisée en veillant à ce que le CUEtotal des cellules affectées par la redistribution soit inférieur à 85% .

Le tableau 3 montre pour 5 scénarios VoIP le pourcentage d'utilisation du canal CUEtotal obtenu en sommant les différents flots résultant de différentes distributions de clients à 11, 5.5, 2 et 1 Mpbs sous 802.11b. Un codec G711 avec 20 ms comme temps de paquétisation est utilisé pour le codage de la VoIP. Le seuil de saturation Th_{alba} vaut 0,85. Le premier scenario considère 1 client fonctionnant à 2 Mbps et 2 clients à 1 Mbps. La charge nécessaire est 0,7 < 0,85. La ressource non utilisée vaut donc 0,85 - 0,7 = 0,15. Comme le montre les scénarios 2 et 3 suivants, seul un nouveau client à 5.5 ou 11 Mbps peut être ajouté à la configuration initiale sans déclencher ALBA. En revanche si 2 nouveaux clients se connectent à 11 Mbps ou si 1 client se connecte à 11 Mbps et 1 autre à 5.5 Mbps, la cellule devient surchargée, comme illustré dans les scenarios 4 et 5. Par conséquent, le seuil Th_{alba} étant franchit une alarme est déclenchée pour lancer l'exécution d'ALBA.

Tableau 3. *Exemple de CUEtotal selon la distribution de clients à 11, 5.5, 2 et 1 Mpbs*

Vitesse (bps)	# clients (cas 1)	# clients (cas 2)	# clients (cas 3)	# clients (cas 4)	# clients (cas 5)
11*10^6	0	1	0	2	1
5.5*10^6	0	0	1	0	1
2*10^6	1	1	1	1	1
1*10^6	2	2	2	2	2
CUEtotal	0,7 < 0,85	0,782 < 0,85	0,801 < 0,85	0,863 > 0,85	0,882 > 0,85

4.2. *Description générique d'ALBA*

On désigne par AP(i) les APs situés au *ième* saut de l'AP surchargé. AP(0) désigne donc l'AP surchargé, AP(1) les APs voisins distants de 1 saut d'AP(0), etc. On appelle périmètre(i) la surface occupée par les AP(i) et zone(i) la surface groupant tous les périmètres (k), $k \leq i$. La description générale du fonctionnement d'ALBA est

représentée par les figures 2 et 3. Les noeuds rouges indiquent les APs à décharger du périmètre (i) et ceux en verts les APs à charger dans le périmètre (i+1).

Etapes principales

Le but d'ALBA est de garantir dans une zone de taille (i) que la charge de chaque AP ne dépasse pas le seuil Th_{alba} pour ne pas affecter la QoS des sessions actives.

Les principales étapes opérationnelles d'ALBA sont les suivantes :

1- ALBA se déclenche au niveau de la zone (0) quand le seuil Th_{alba} est franchit.

2- Si l'algorithme n'arrive pas à décharger la zone (i), la taille de la zone est incrémentée de 1 (cf figure 2). La taille maximale de la zone de décharge est fixé à n_{max}. La décharge réussie si elle permet d'insérer la nouvelle station.

3- La décharge d'une zone i commence par la décharge des stations du périmètre (i) au périmètre (i+1), est exécutée de proche en proche, et se termine par la décharge du périmètre (0) au périmètre (1) (voir figure 3). Selon cette procédure, la réassociation se fait progressivement par une propagation centrifuge des stations.

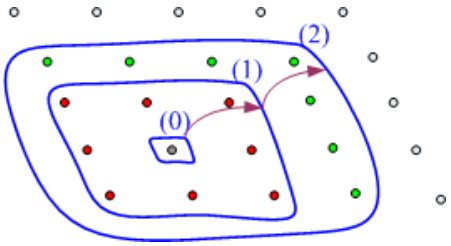


Figure 2. *Etape 2 : Agrandissement de la zone de décharge*

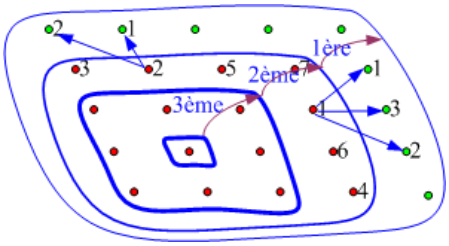


Figure 3. *Etape 3 : Déplacement centrifuge des stations*

Stratégie de décharge

La politique de sélection des stations à réassocier, des AP(i) à décharger et des AP(i+1) à charger représente l'aspect majeur de l'opération de décharge (étape 3).

a. Les AP(i) à décharger sont choisis dans l'ordre décroissant de charge.

b. Les AP(i+1) à charger sont choisis dans l'ordre croissant de charge.

c. Les stations les plus lentes (1Mbps) sont choisies en premier, les plus rapides en dernier (11Mbps). Cet ordre s'explique par le fait que les stations les plus éloignées sont celles dont le CUE peut être le mieux réduit. Ces stations sont aussi celles contribuant le plus au CUE. La réassociation d'une station est possible au niveau d'un AP du périmètre (i+1) si l'estimation du CUE de la station ajoutée au CUE des flots déjà connectés à l'AP est inférieure à Th_{alba}.

Une fois le plan de décharge validé et établi dans l'AP(0) celui-ci est mis en oeuvre dans le réseau sous la forme d'une transaction atomique. Pour ce faire, l'AP désigné pour chaque WD est communiqué aux stations qui se réassocient immédiatement, en même temps.

5. Simulations

Les simulations sont réalisées sur un simulateur intégrant le modèle de CUE présenté à la section 4.1 et tel que décrit dans [GAR 03b]. Les résultats obtenus étendent ceux des expériences préliminaires réalisées sous le simulateur de paquets Opnet [SAW 08].

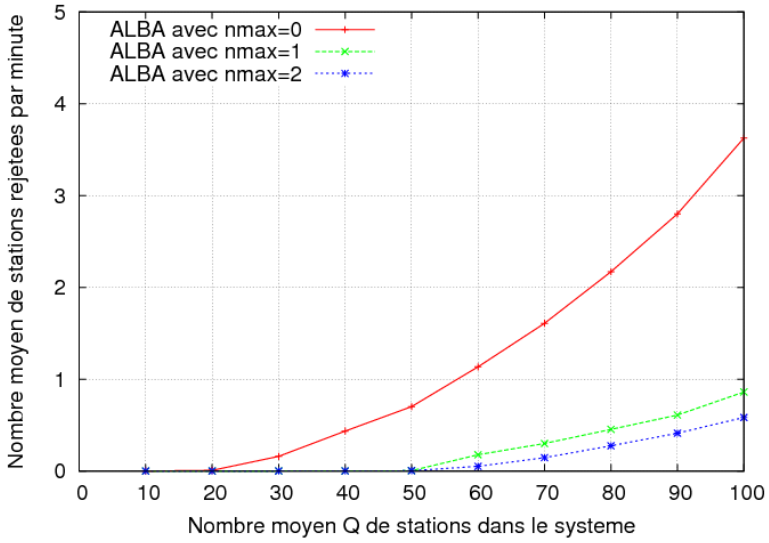


Figure 4. *Nombre moyen de stations rejetées par minute*

5.1. *Description du système*

La surface du terrain est de 1500 x 1500 m^2. Le coeur du terrain est 1-couvert à 11 Mbps par 6 x 6 APs 802.11b disposés en grille. Les portées à 1, 2, 5.5 et 11 Mbps sont celles d'une carte Cisco 340 et valent respectivement 121, 243, 365 et 457 m. Dans ces conditions de déploiement des APs, le terrain est 9-couvert à 1 Mbps. Les stations arrivent dans la zone de couverture selon une loi de Poisson de paramètres λ. La durée de chaque session suit une loi exponentielle de moyenne μ^{-1} (5 min/appel). Les positions des stations en x et y sont générées selon une loi normale ayant respectivement comme espérance et écart type le centre et la moitié du terrain. En régime stationnaire, le nombre moyen de client Q vaut $\lambda * \mu^{-1}$. La valeur de λ fixe donc le nombre de stations Q présentes simultanément sur le terrain. Les stations sont supposées immobiles. Chaque station génère à son arrivée un appel VoIP. Le codec G711 est utilisé pour le codage de la voix. Les trames de données audio sont envoyées chaque 20 ms et contiennent 160 octets de charge utile, sans compression.

Le trafic VoIP circule entre les stations et les APs. Le système de distribution est connecté à l'internet via un lien très haut débit. Seule la période du régime station-

naire est conservée dans les résultats. La figure 4 montre le nombre moyen de stations rejetées par minute durant la simulation. Les résultats sont les moyennes de 10 simulations. Les simulations diffèrent par les coordonnées x et y des stations.

5.2. *Résultats des simulations sans algorithme de répartition de charge*

Dans ce cas, aucun algorithme de répartition de charge n'est appliqué et n_{max} vaut 0. Les stations se connectent toujours à l'AP le plus proche. Le problème se ramène à un algorithme de contrôle d'admission classique où les stations sont rejetées si le seuil de CUE est dépassé. Aucune gestion n'étant réalisée, les performances sont moins bonnes que lorsqu'ALBA est activé. Le nombre de stations rejetées croit quasi linéairement pour atteindre un rythme de 4 stations par minute lorsque 100 stations sont présentes simultanément sur le terrain. En d'autre terme, lorsque les stations arrivent selon un rythme λ de 20 stations par minute, 4 stations seront rejetées, soit un taux de rejet de 20% environ.

5.3. *Résultats des simulations avec ALBA*

Lorsqu'ALBA est appliqué (n_{max}=1 et 2), les APs au voisinage de l'AP surchargé peuvent être reconfigurés pour résoudre le problème de la congestion. La figure 4 montre que les premières pertes apparaissent à 50 stations avec n_{max}=1 et à 60 stations avec n_{max}=2. ALBA avec n_{max}=2 fournit en moyenne des performances 50% supérieures en terme de pourcentage de stations rejetées comparé à ALBA avec $n_{max} = 1$. Ce gain s'explique par le fait que la capacité d'ALBA à évacuer la congestion en bordure des zones surchargées de stations croit avec n_{max}.

6. Conclusion

Nous avons proposé dans cet article un système de répartition de charge autonomique afin d'optimiser l'utilisation de la capacité dans les réseaux WLAN. Une architecture de système basée sur le Plan de Connaissance et le concept de la vue située est proposée à ce titre.

L'étude de performance montre des gains substantiels en terme de capacité d'accueil en stations du réseau avec ALBA à 1 saut. Ces gains augmentent encore de près de 50% à 2 sauts. Ceci confirme l'intérêt de dégrader la vitesse physique des équipements pour désengorger le coeur du réseau si les conditions de CUE le permettent.

Comme travaux futurs, nous envisageons d'étendre ALBA avec une gestion proactive du réseau destinée à l'optimisation de l'équilibrage de la charge. La deuxième perspective consiste à intégrer d'autres paramètres de décision de répartition de charge tels que les profils d'utilisateurs.

7. Bibliographie

[BAL 02a] BALACHANDRAN A., BAHL P., VOELKER G., « Hot-spot congestion relief in public-area wireless networks », *4th IEEE Workshop on Mobile Computing Systems and Applications 2002*, Callicoon, USA, 2002.

[BAL 02b] BALACHANDRAN A., VOELKER G., BAHL P., RANGAN V., « Characterizing User Behavior and Network Performance in a Public Wireless LAN », *ACM SIGMETRICS Performance Evaluation Review 2002*, Marina Del Rey, Californie, USA, 2002.

[BAL 03] BALAZINSKA M., CASTRO P., « Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network », *Proceedings of MobiSys 2003*, San Francisco, CA, USA, 2003.

[BEJ 04] BEJERANO Y., HAN S.-J., LI L., « Fairness and Load Balancing in Wireless LANs Using Association Control », *MobiCom 2004*, Philadelphia, PA, September 2004.

[CIS 04] CISCO, « Data sheet for cisco aironet 1200 series », Technical report, 2004, Cisco Systems Inc, INRIA.

[CLA 03] CLARK D. D., PARTRIDGE C., RAMMING J. C., WROCLAWSKI J. T., « A Knowledge Plane for the Internet », *SIGCOMM 2003*, Karlsruhe, Germany, 2003.

[DAH 06] DAHER R., TAVANGARIAN D., « QoS-Oriented Load Balancing for WLANs », *First IEEE workshop OpCommin, 2006*, Berlin, Germany, 2006.

[GAR 03a] GARG S., KAPPES M., « Admission Control for VoIP Traffic in IEEE 802.11 Networks », *GLOBECOM 2003*, San Francisco, USA, 2003.

[GAR 03b] GARG S., KAPPES M., « Can I add a VoIP call ? », *IEEE ICC 2003*, Anchorage, Alaska, 2003.

[GAR 03c] GARG S., KAPPES M., « An Experimental Study of Throughput for UDP and VoIP Traffic in IEEE 802.11b Networks », *IEEE WCNC 2003*, New Orleans, LA, 2003.

[HEU 03] HEUSSE M., ROUSSEAU F., SABBATEL G. B., DUDA A., « Performance Anomaly of 802.11b », *IEEE INFOCOM 2003*, San Francisco, USA, 2003.

[IEE 97] IEEE802.11, « IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications », Ieee std 802.11, 1997.

[SAW 08] SAWMA G., AIB I., BEN-EL-KEZADRI R., PUJOLLE G., « ALBA : An Autonomic Load Balancing Algorithm for IEEE 802.11 Wireless Networks », *à paraître, papier court, NOMS 2008*, Salvador, Bahia, Brésil, Avril 2008.

[SHE 99] SHEU S.-T., WU C.-C., « Dynamic Load Balance algorithm (DLBA) for IEEE 802.11 Wireless LAN », *Tamkang Journal of Science and Engineering*, , 1999.

[STR 06] STRASSNER J., « Autonomic Networks and systems - Theory and Practice », *Tutorial NOMS 2006*, Vancouver, Canada, 2006.

[VEL 04] VELAYOS H., ALEO V., KARLSSON G., « Load balancing in overlapping wireless LAN cells », *IEEE ICC 2004*, Paris, France, 2004.

Posters

Prototype de canal caché dans le DNS

Lucas Nussbaum et Olivier Richard

Laboratoire d'Informatique de Grenoble - projet MESCAL
Antenne de Montbonnot
ZIRST 51, avenue Jean Kuntzmann
38330 Montbonnot Saint Martin
{lucas.nussbaum,olivier.richard}@imag.fr

RÉSUMÉ. *Avec la multiplication des réseaux ne permettant qu'un accès restreint à Internet, de nombreuses techniques ont vu le jour pour s'évader de ces restrictions. Les canaux cachés dans le DNS sont l'une d'elles, consistant à utiliser des requêtes et des réponses DNS pour transmettre et recevoir des informations via un serveur DNS complice. Après une présentation des différentes solutions existantes, nous présentons notre prototype, TUNS, et nous l'évaluons en le comparant aux autres implantations.*

ABSTRACT. *With more and more networks providing a restricted access to the Internet, numerous solutions have been developed to evade such networks. Covert channels inside DNS, which leverage DNS requests and replies to transmit and receive data using a rogue DNS server, are one of them. After describing the varous existing solutions, we introduce our own prototype, TUNS, and evaluate it by comparing it to other implementations.*

MOTS-CLÉS : *canal caché, DNS, mesure de performances*

KEYWORDS: *covert channel, DNS, performance evaluation*

1. Introduction

Avec la multiplication des réseaux ne permettant qu'un accès partiel à Internet (réseaux d'entreprises, d'hôtels, de restaurants), de nombreuses personnes ont essayé d'utiliser les quelques protocoles disponibles pour obtenir un accès complet à Internet. Cela se fait en général en transmettant les données en utilisant un protocole autorisé, vers un service complice situé ailleurs sur Internet, sur un réseau non limité.

Ainsi, il existe des implantations de tunnels utilisant les protocoles HTTP, HTTPS, ou ICMP. Dans ce poster, nous nous intéressons à l'utilisation du protocole DNS dans ce cadre.

Le protocole DNS est disponible sur de nombreux réseaux restreints, car il est nécessaire pour permettre d'utiliser la plupart des autres protocoles. Seuls les réseaux ne permettant qu'un accès au protocole HTTP peuvent se permettre de ne pas offrir un accès au protocole DNS : dans ce cas, la résolution DNS se fait sur le serveur proxy.

Toutefois, l'encapsulation de données dans des requêtes et des réponses DNS est difficile. Dans les requêtes, la seule possibilité est d'encoder les données dans le nom à résoudre, en codant les données sous une forme textuelle (Base32, ou Base64, en prenant quelques libertés avec la RFC 1035). Dans les réponses, les enregistrements TXT et NULL permettent d'envoyer de grosses quantités d'informations, mais ils sont souvent filtrés par les serveurs DNS. Une extension du protocole DNS définie dans la RFC 2671 permet aussi d'augmenter la taille des réponses.

Dans la suite de cet article, nous présenterons quelques implantations existantes de canaux cachés utilisant le DNS. Puis nous présenterons notre prototype, TUNS, ainsi que les résultats d'évaluations comparant TUNS aux autres implantations.

2. Implantations existantes

On peut distinguer deux catégories d'inplantations :

– Les solutions qui fournissent une connectivité au niveau IP (tunnels IP over DNS) ;

– Les solutions qui fournissent un seul canal de communication, permettant par exemple d'établir une connexion SSH.

Les tunnels fournissant une connectivité au niveau IP fournissent un périphérique `tun` ou `tap`, permettant à l'utilisateur de router les paquets souhaités à travers le tunnel. Leur utilisation est donc totalement transparente pour les applications.

NSTX [nst] est historiquement le plus ancien. Pour encoder les données dans les requêtes, il utilise un encodage Base64 non-conforme à la RFC 1035 (il utilise des caractères "_" en plus des 63 caractères autorisés par la RFC. Les réponses sont contenues dans des paquets TXT.

Iodine [iod] est un projet plus récent. Il utilise (au choix) un encodage Base32 ou Base64 pour encoder les données dans les requêtes, et des paquets de type NULL pour les réponses. Il utilise également EDNS0 pour augmenter la taille des réponses.

À la fois NSTX et Iodine découpent les paquets IP à envoyer en plusieurs paquets DNS, les envoyent séparément, et recomposent les paquets IP côté serveur.

La deuxième catégorie de tunnels ne fournit qu'une connexion TCP. L'utilisateur établit en général une connexion SSH, puis utilise les fonctionnalités de *port forwarding* et de proxy SOCKS de SSH.

La difficulté de ces solutions est qu'elles doivent fournir un canal fiable au dessus d'un protocole non fiable, et donc gérer les pertes, réordonnancements et duplications de paquets DNS.

OzymanDNS [ozy] est la solution la plus utilisée. Il utilise des enregistrements TXT, et l'extension EDNS0. Nous avons constaté des plantages très fréquents lors de nos tests.

dns2tcp [dns] est une autre solution. Il utilise des enregistrements TXT et un encodage Base64 non-conforme (utilisation de '/').

3. TUNS

Après une étude des solutions existantes, nous avons proposé notre propre solution, avec pour objectif de se restreindre à un fonctionnement le plus standard possible (et donc le plus difficile à empêcher). TUNS est un tunnel IP over DNS, écrit en Ruby. Contrairement aux autres solutions, qui utilisent des enregistrements TXT ou NULL, rarement utilisés de manière légitime, TUNS n'utilise que des enregistrements CNAME. Pour encoder les données émises côté client et serveur, un encodage Base32 est utilisé. Contrairement à NSTX et Iodine, TUNS ne découpe pas les paquets IP entre plusieurs paquets DNS plus petits : à la place, le MTU de l'interface du tunnel est réduit, et c'est donc le système d'exploitation qui se charge du découpage en utilisant la fragmentation IP.

Lorsque des données à transmettre sont reçues par TUNS côté client, elles sont immédiatement transmises dans une requête DNS.

Pour recevoir des données du serveur, le client sonde régulièrement le serveur avec des requêtes quasi-vides. Le serveur répond immédiatement à ces requêtes si des données à transmettre au client sont disponibles. Si ce n'est pas le cas, il attend quelques dixièmes de seconde, au cas où des données à transmettre arriveraient pendant cette attente.

4. Évaluation

Nous avons évalué Iodine, NSTX et TUNS.

Tout d'abord, nous avons vérifié le bon fonctionnement des trois solutions sur différents réseaux. Il est apparu que Iodine et NSTX n'étaient pas utilisable sur de nombreux réseaux, probablement à cause des libertés prises par rapport à la RFC 1035. Par contre, nous n'avons pas trouvé de réseau sur lequel TUNS ne fonctionnait pas.

Dans un deuxième temps, nous nous sommes intéressés aux performances de ces 3 outils. Nous avons réalisé des expériences à l'aide d'un émulateur réseau, permettant de modifier la latence entre le client et le serveur. Trois machines de Grid'5000 ont été utilisées avec l'émulateur TC+Netem inclu dans Linux.

Nous avons d'abord mesuré le *Round Trip Time* à l'aide de *pings* envoyés à travers le tunnel, initiés par le serveur. Si NSTX et TUNS ne modifient pas significativement la latence (la latence mesurée à travers le tunnel est proche de la latence physique), on constate (figure 1) que Iodine provoque une latence bien plus importante. Cela est probablement dû à l'algorithme de sondage utilisé par le client.

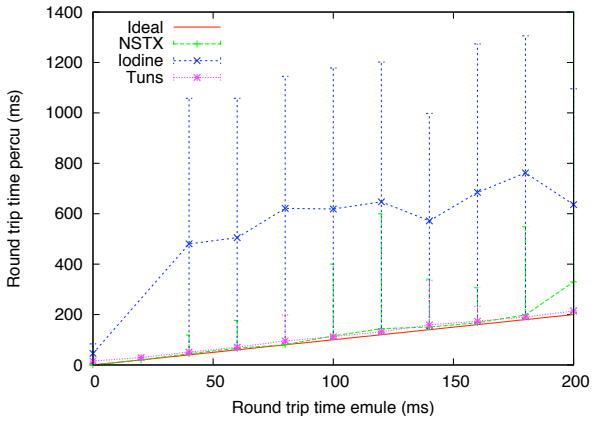


Figure 1. *Round Trip Time mesuré à travers tunnel, comparé au RTT réel entre le client et le serveur*

Puis nous nous sommes intéressés au débit maximal disponible à travers le tunnel. Encore une fois, nous l'avons mesuré en émulant une latence plus importante entre le client et le serveur. À l'aide de l'émulateur, nous avons également émulé 2% de pertes de paquets. Le comportement des tunnels face à des pertes de paquets est important, puisqu'ils sont souvent utilisés sur des réseaux WiFi de mauvaise qualité. Nous constatons (figure 2) que les performances de TUNS sont inférieures à celles de Iodine et NSTX. Mais face à des pertes de paquets, TUNS est moins affecté que les deux autres solutions, probablement car il ne découpe pas les paquets IP en plusieurs paquets DNS (la perte d'un paquet DNS ayant alors des conséquences plus importantes).

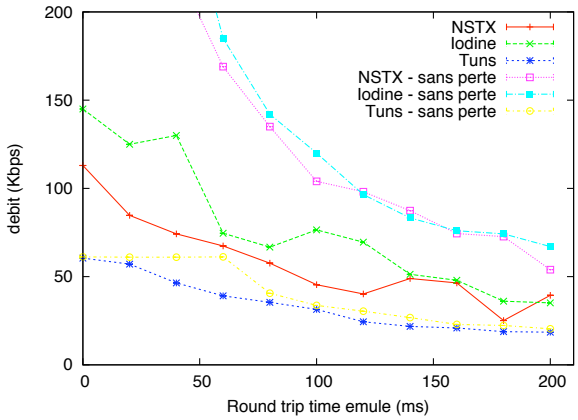


Figure 2. *Débit montant (client vers serveur) disponible à travers le tunnel, en fonction du RTT réel entre le client et serveur*

5. Conclusion

Après une présentation des différentes solutions existantes permettant de réaliser un canal d'informations caché dans des requêtes DNS, nous avons proposé notre propre prototype, TUNS, conçu dans l'objectif d'être le plus conforme possible au protocole DNS, et nous l'avons évalué en le comparant aux autres implantations existantes.

En terme de latence, nous avons constaté que TUNS proposait des performances équivalentes à celles des autres solutions. Toutefois, en terme de débit, ses performances sont moins bonnes. Cela s'explique, d'une part, par l'absence de découpage des paquets IP : l'utilisation de la fragmentation diminue la part utile de données dans les paquets DNS. D'autre part, nous avons constaté une importante utilisation du processeur dans nos expériences : il semblerait que l'implantation en utilisant un langage de script, en particulier de la bibliothèque gérant la construction et le décodage des paquets DNS, est assez peu performante.

6. Bibliographie

[dns] « Dns2tcp », `http://hsc.fr/ressources/outils/dns2tcp/`.

[iod] « Iodine », `http://code.kryo.se/iodine/`.

[nst] « NSTX », `http://thomer.com/howtos/nstx.html`.

[ozy] « OzymanDNS », `http://www.doxpara.com/`.

Poster : Propriétés structurelles des réseaux ad hoc de véhicules

Florent Kaisser, Anthony Busson, Véronique Vèque

Institut d'Electronique Fondamentale, Bat 220 University Paris Sud, Centre Scientific Orsay, 91405 cedex France email : firstname.lastname@ief.u-psud.fr

RÉSUMÉ. *Un réseau ad hoc de véhicules ou VANET est constitué de véhicules capables de s'échanger des informations par voie radio dans le but d'améliorer la sécurité routière ou de permettre l'accès à Internet pour les passagers, que nous avons choisi comme scénario. Le VANET se différencie par une forte mobilité des noeuds rendant la topologie du réseau fortement dynamique. Cette étude structurelle des réseaux de véhicules, vise à étudier la dynamique du trafic routier grâce à un modèle analytique et par une simulation. Nous montrons que dans un cadre VANET, DSR est meilleur que GPSR pour le taux de réception. Nous en déduisons également, qu'un réseau de véhicules purement Ad Hoc n'est pas viable pour les applications utilisateurs et concluons sur la nécessité de le coupler à une infrastructure ou réseau hybride.*

ABSTRACT.

MOTS-CLÉS : *réseau ad hoc, VANET, véhicule, hybride, DSR, GPSR, performance, NS-2, trafic routier, capacité*

KEYWORDS: *ad hoc network, VANET, vehicular, hybrid, DSR, GPSR, performance, NS-2, road traffic, capacity*

1. Les réseaux ad hoc de véhicules

Un réseau ad hoc de véhicules ou VANET est constitué de véhicules capables de s'échanger des informations par voie radio dans le but d'améliorer la sécurité routière ou de permettre l'accès à Internet pour les passagers, que nous avons choisi comme scénario. Par rapport à un réseau ad hoc classique, le VANET se différencie par une forte mobilité des noeuds rendant la topologie du réseau fortement dynamique. Les réseaux mobiles actuels peinent à satisfaire des communications haut débit à grande vitesse. Ainsi, les versions les plus évoluées de UMTS, HSPA (high-speed packet access), annoncent des débits maximals de 3.6 Mbit/s mais qui en moyenne seront plutôt de l'ordre de 750 kbit/s et principalement sur le lien descendant. La mise en place de réseaux de communication inter-véhicules ne fera donc pas double emploi avec les réseaux mobile d'opérateurs. La première façon de concevoir un tel réseau est de limiter les déploiements et de la faire à un coût léger sous la forme d'un réseau Ad Hoc de véhicules. Nous avons ainsi abordé deux aspects : le premier portant sur les propriétés topologiques du réseau ad hoc et le second, sur son aptitude à transporter des données d'une application. Nous avons cherché à répondre aux questions suivantes :

– à quoi ressemble la topologie du réseau VANET ?

– les communications pour l'accès à Internet sont-elles possibles ?

– quelle est l'influence du mouvement des véhicules sur le routage ?

– quelle est la capacité ou le débit disponible dans un tel réseau sans fil ?

aux possibilités de communications entre véhicules qui en découlent.

2. Modèle analytique et simulation

Pour mener notre étude structurelle des réseaux de véhicules, nous avons étudié la dynamique du trafic routier et nous nous sommes basés à la fois sur un modèle analytique et sur un simulateur. Un simulateur du modèle microscopique du trafic routier a été développé en C pour s'interfacer avec NS-2. Ce simulateur consiste, pour chaque noeud, à reproduire le comportement type d'un automobiliste qui réagit en fonction de son environnement [AHM 99]. A chaque véhicule est associée une vitesse cible qui est la vitesse à laquelle le conducteur roulerait s'il était seul sur sa voie. S'il n'est pas seul, trois types de comportement sont identifiés :

– Free-flow : le véhicule se déplace librement et peut changer de voie sur tirage d'une probabilité

– Car-following : un véhicule se déplace en fonction du véhicule qu'il suit sur la même voie ;

– Emergency : le véhicule maintient une distance de sécurité avec le véhicule qui est devant lui.

Dans un contexte d'autoroute, le trafic routier, et notamment chaque noeud, ont pu être modélisés par un Processus Ponctuel de Poisson. Grâce à ces deux modèles,

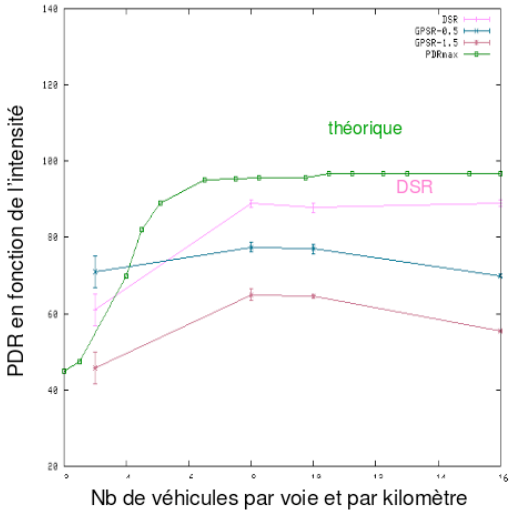


Figure 1. *Comparaison DSR/GPSR sur le taux de perte*

nous avons montré la fragmentation du réseau et la formation de grappes de véhicules isolées les unes des autres [MAB 06, M.M 07]. Le modèle analytique permet de calculer la taille moyenne d'une grappe, mais aussi sa longueur ainsi que la probabilité que deux véhicules soient connectés (c'est-à-dire à une distance inférieure à la portée radio) et la durée de cette connexion. Sans surprise, cette probabilité diminue avec la distance et augmente avec le nombre de véhicules sur l'autoroute. L'étude par simulation des délais de connexions puis de reconnexion montre de plus qu'une grande majorité des connexions durent moins d'une seconde et la déconnexion qui suit est trop longue pour permettre ne serait-ce qu'un accès Web.

3. Comparaison des protocoles de routages

Le simulateur du trafic routier interfacé avec NS-2 nous a ensuite permis de comparer deux algorithmes de routage, parmi les plus répandus, DSR (Dynamic Source Routing [JOH 96]) et GPSR Greedy Perimeter Stateless Routing) [KAR 00]. Le choix de ces deux protocoles est justifié par le fait que nous voulions comparer un protocole réactif avec un protocole proactif basé sur une fonction de localisation, dans l'hypothèse où dans un futur proche tous les véhicules seraient équipés de terminaux GPS. Les protocoles proactifs tel que OLSR ne supportant pas la forte dynamicité du réseau, ils ne seront pas inclus dans notre comparaison. Notre étude affine les précédents travaux sur le routage car le modèle de trafic est plus réaliste que les modèles de mobilité aléatoires généralement considérés. Nous avons ainsi caractérisé le taux de paquets reçus pour chacun des protocoles. Dans les deux cas, la vitesse joue un rôle important.

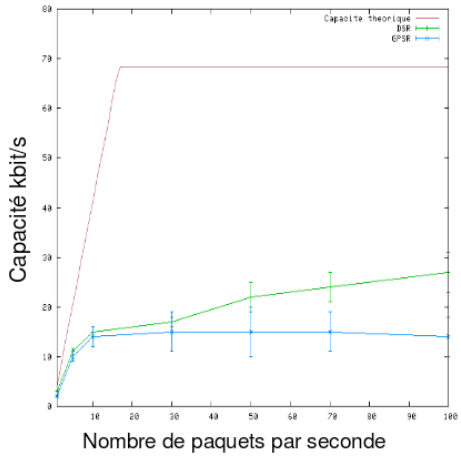


Figure 2. *Etude de la capacité Variation du débit/source*

Pour DSR, le délai de découverte de chemins par une source peut induire des pertes si un des noeuds est hors de portée de ses voisins sur le chemin. Pour GPSR, on a le même problème avec une liste de voisins devenus obsolète. Ainsi, le taux de réception de paquets est meilleur si le nombre de véhicules par km est plus grand ce qui a été confirmé par l'étude analytique et la courbe des probabilités d'existence d'une connexion. Si le débit des connexions augmente, le taux de réception diminue comme nous le démontrons lors de l'étude de la capacité. Enfin, la propriété d'overhead de routage est importante car les paquets de contrôle occupe aussi de la bande passante et là encore, DSR est plus performant. En conclusion et contrairement à des publications antérieures [F¨ 03, Füß 02] où les conditions étudiées étaient particulièrement favorables, nous montrons que dans un cadre VANET, DSR est meilleur que GPSR à tous points de vue (figure 1) .

Enfin, nous avons étudié la capacité disponible en nous basant sur le modèle de chaîne d'émetteurs [LI 01].

4. Capacité du réseau

La capacité disponible est défini ici comme le débit de réception des paquets au niveau d'un noeud. Dans une telle topologie en ligne et avec des émetteurs fixes, un noeud relais doit retransmettre un paquet à son voisin qui fait de même jusqu'à ce que le paquet arrive à destination. Chaque noeud relais reçoit donc un paquet deux fois et le retransmet une fois. Cela diminue le taux d'utilisation du canal de $\frac{1}{3}$. De plus, les interférences ont un plus grand rayon que les transmissions correctes (par exemple,

550 mètres contre 250 mètres) ce qui diminue d'encore $\frac{1}{4}$ la capacité disponible. Finalement, la capacité est affectée par le protocole avec contention en fonction du nombre de noeuds émetteurs à un instant donné. La capacité trouvée dans [LI 01] n'est alors plus que de $\frac{1}{8}$ du débit brut et nous l'avons utilisée comme borne théorique. Nos simulations qui prennent en compte à la fois la mobilité et le routage montre que la capacité obtenue est inférieure à la moitié de la borne théorique (figure 2) issue du modèle de la chaîne d'émetteurs [MAB 07].

Nous en déduisons qu'un réseau de véhicules purement Ad Hoc n'est pas viable pour les applications utilisateurs et concluons sur la nécessité de le coupler à une infrastructure ou réseau hybride. Une première étude théorique du réseau hybride nous a permis de dimensionner le nombre de points d'accès en fonction des caractéristiques connues du trafic routier (intensité, vitesse). Notre recherche actuelle porte sur la mise en place d'un protocole de routage pour les réseaux hybrides efficace .

5. Bibliographie

[AHM 99] AHMED K. I., « Modeling Drivers'Acceleration and Lane Changing Behavior », Doctor of Science in Transportation Systems, Massachusetts Institute of Technology, Boston (MA), 1999.

[F¨ 03] FÜSSLER H., WIDMER J., KÄSEMANN M., MAUVE M., HARTENSTEIN H., « Beaconless Position-Based Routing for Mobile Ad-Hoc Networks », rapport n° TR-03-001, February 2003, Department of Computer Science, University of Mannheim.

[Füß 02] FÜSSLER H., MAUVE M., HARTENSTEIN H., KÄSEMANN M., VOLLMER D., « A Comparison of Routing Strategies for Vehicular Ad Hoc Networks », rapport n° TR-02-003, July 2002, Department of Computer Science, University of Mannheim.

[JOH 96] JOHNSON D. B., MALTZ D. A., BROCH J., « Dynamic Source Routing in Ad Hoc Network », *Mobile Computing, Kluwer Academic Publishers*, 1996, p. 153-181.

[KAR 00] KARP B., KUNG H. T., « GPSR : Greedy Perimeter Stateless Routing for Wireless Networks », *The sixth Annual ACM/IEEE International Conference On Mobile and Networking (MobiCom 2000)*, Aout 2000.

[LI 01] LI J., BLAKE C., COUTO D. S. D., LEE H. I., MORRIS R., « Capacity of Ad Hoc wireless networks », *MobiCom '01 : Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, NY, USA, 2001, ACM Press, p. 61–69.

[MAB 06] MABIALA M., BUSSON A., VÈQUE V., « Analyse du trafic et du routage dans un réseau Ad Hoc de véhicules », *Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006*, 2006.

[MAB 07] MABIALA M., BUSSON A., VÈQUE V., « On the capacity of Vehicular ad hoc Networks », *Workshop UBIROADS 2007*, 2007.

[M.M 07] M. MABIALA A. B., VÈQUE V., « Inside VANET : hybrid network dimensionning and routing protocol comparison », *IEEE 65th Vehicular Technology Conference VTC2007-Spring*, April 2007.

Un Algorithme de Gestion des Adjacences basé sur la Puissance du Signal[1]

Husnain Mansoor Ali — Anthony Busson — Amina Meraihi Naimi — Veronique Veque

University Paris XI
IEF - CNRS UMR 8622
Centre Scientifique d'Orsay
91405 Orsay - France

{husnain.ali,anthony.busson,amina.meraihi,veronique.veque}@ief.u-psud.fr

RÉSUMÉ. *Dans cette proposition, nous présentons une technique de gestion de lien pour les protocoles proactifs de routage dans les réseaux ad-hoc. Ce nouveau mécanisme de type multi-couche est basé sur la puissance du signal. Le mécanisme d'hystérésis prévu par le protocole OLSR est amélioré par l'utilisation de la puissance du signal combinée avec les pertes de messages Hello. La puissance du signal est utilisée pour déterminer si la qualite de lien s'améliore ou est en baisse. Cela nous aide, non seulement à rendre la gestion de liens plus robuste, mais également à anticiper la rupture du lien et donc à améliorer les performances.*

ABSTRACT. *In this proposition, we present a link management technique for pro-active routing protocols for ad-hoc networks. This new mechanism is based on signal strength hence cross layer approach is used. The hysteresis mechanism provided by OLSR is improved upon by using signal strength in combination with the hello loss based hysteresis. The signal power is used to determine if the link-quality is improving or deteriorating while packet losses are handled through the hysteresis mechanism specified in OLSR RFC. This not only makes the link management more robust but also helps in anticipating link breakages thereby greatly improving the performance.*

MOTS-CLÉS : *Protocoles de Routage, Gestion de Lien, OLSR, MANET, Puissance du Signal*

KEYWORDS: *Routing Protocols,Link Management, OLSR, MANET, Signal Strength*

1. This work is supported by the French government funded project ANR RNRT R2M (Reseaux Mesh et Mobilite)

1. Introduction

Dans les réseaux sans-fil, la mobilité des noeuds cause la rupture des liens radios. Cela peut avoir pour conséquences une dégradation importante des performances surtout en présence de mobilité rapide. Dans les réseaux ad-hoc, la gestion des adjacences, c'est-à-dire l'établissement de la liste des autres noeuds à portée radio est réalisé localement par chaque noeud. La gestion de ces adjacences est effectué par le protocole de routage. Cette tâche est généralement effectué sur la base de la réception ou non réception de messages de contrôles (message `Hello`). Donc habituellement, la puissance du signal pour la réception des messages `Hello` n'est pas considéré, en effet la couche réseau fait très peu d'hypothèse quant à la technologie sous-jacente. Mais cela peut entraîner des taux de perte important [RAI], lorsque le wifi est utilisé notemment. Aussi il existe quelques approches où la puissance du signal est prise en compte, mais cette approche est utilisé principalement dans le contexte des protocoles de routage réactifs [CRI 04]. Dans cet article, nous nous focalisons plutôt sur les protocoles de routage proactifs. Avec un protocole proactif, les tables de routage contiennent une entrée pour chaque destination dans le réseau. Il ne nécessite donc pas de requêtes pour déterminer la route lors d'une nouvelle communication.

Dans cet article, nous proposons un algorithme efficace de gestion des adjacences dans le cadre d'une mobilité rapide (au delà de 60km/h) prenant en compte la puissance du signal. On suppose de plus qu'il s'agit d'un réseau mesh. Dans ce contexte, un réseau mesh est un réseau ad hoc où un certain nombre de noeuds, en principe statique, ont été déployés pour effectuer le relayage, ces noeuds accroissent la connectivité du réseau ad hoc et offrent plusieurs chemins possibles entre les sources et les destinations.

Notre algorithme est une modification de l'algorithme de gestion des adjacences d'OLSR (Optimized Link State Routing Protocol) [Cla 03]. Cependant, il peut être utilisé par n'importe quel protocole de routage. Le problème avec l'algorithme initial est qu'il ne prend en compte que le fait que les paquets soient reçus ou perdus. Pour éviter de dévalider un lien à la moindre perte de message `Hello`, qui peut être provoqué par des collisions ou un évanouissement radio important, il y a un mécanisme d'hystérésis qui fait qu'il faut 2 pertes de `Hello` succéssives (avec les paramètres par défaut) pour considérer un lien invalide. Cet hystérésis permet de stabiliser les liens radio, et est donc nécessaire pour palier aux pertes dûes à des phénomènes transitoires (collisions, evanouissement de la puissance radio). Mais lorsque deux noeuds s'éloignent l'un de l'autre, il faudra environ 4 secondes avant de considérer le lien invalide. Ces 4 secondes correspondent au temps entre le moment où les noeuds ne sont plus à portée radio et l'émission de deux `Hello` (qui ne sont pas reçus). Durant cette période, les deux noeuds considèrent le lien valide et peuvent émettre des données sur celui-ci. Toutes les données émises seront alors perdues. L'amélioration que nous apportons au mécanisme natif d'OLSR est la prise en compte de la détéroriation ou amélioration du signal. Nous n'attendons pas la perte sèche d'un ou deux `Hello` pour dévalider un lien, mais nous le dévalidons si durant plusieurs `Hello` succéssifs il y a une perte signicative de la puissance du signal.

Dans la Section suivante, nous présentons notre algorithme, puis Section 2.1 nous donnons quelques résultats de simulations. Nous concluons Section 3.

2. Algorithme de gestion des adjacences

Comme dans OLSR, on associe à chaque lien une métrique de qualité. Cette métrique de qualité est sans unité. Lorsque cette métrique passe en dessous d'un seuil bas le lien n'est plus utilisé, lorsqu'elle passe au dessus d'un seuil haut le lien devient de nouveau utilisable. Dans OLSR, cette métrique est mis à jour uniquement en fonction des messages `Hello` perdus ou reçus. L'idée de notre algorithme est assez simple. Il s'agit d'augmenter la qualité du lien lorsque la qualité du signal radio augmente et inversement. Sous l'hypothèse que la puissance du signal décroit avec la distance, on augmente la métrique du lien lorsqu'un noeud se rapproche et on la diminue quand un noeud s'éloigne de manière à dévalider le lien juste avant que le noeud sorte de la portée radio. On utilise alors deux seuils (*ss_threshold_low* et *ss_threshold_high*) associés à la puissance du signal (et non pas à la métrique du lien). Lorsque la puissance du signal du `Hello` reçu est en dessous du seuil bas (*ss_threshold_low*), on diminue la métrique de qualité. Lorsque la puissance du signal est entre les deux seuils, on est à la limite de la portée radio, et on prend en compte la puissance du signal pour savoir si le noeud s'éloigne ou se rapproche. Si la puissance du signal s'est nettement améliorée on augmente la qualité du lien, si au contraire il y a une détérioration nette on diminue la qualité du lien. Enfin, lorsque la puissance du signal est au dessus du seuil haut (*ss_threshold_high*), on a une bonne qualité de lien radio et on augmente la métrique du lien (qu'il y ait dégradation ou non).

Nous donnons ci-dessous, l'algorithme en détail.

Signal Strength(ss) based Hysteresis Algorithm. .
IF there does not exist an entry in neighbor table
Initialize Link_quality
Link_pending $\leftarrow$ *true*
Sum_sig_var = 0 ; (to accumulate the signal variation)
ELSE (if there is already an entry)
IF ss $>$ *ss_threshold_high ; (good reception ; we reward)*
Link_quality $\leftarrow$ *(1 - Hyst_ss_scaling) * Link_quality + Hyst_ss_scaling*
ELSE (punish, reward or do nothing based on ss)
IF Link_pending = false AND (Sum_sig_var $+= Last_ss - ss) \geq \Delta$
(punish ; signal strength deterioration $\geq \Delta$*)*
Link_quality $= Hyst_ss_scaling * Link_quality$
Sum_sig_var = 0
ENDIF
IF Link_pending = true AND (Sum_sig_var $+= ss - Last_ss) \geq \Delta$
(reward ; signal strength improvement $\geq \Delta$*)*
Link_quality = min(HYST_THRESHOLD_HIGH , $(1-Hyst_ss_scaling)*$
$Link_quality + Hyst_ss_scaling$ *)*

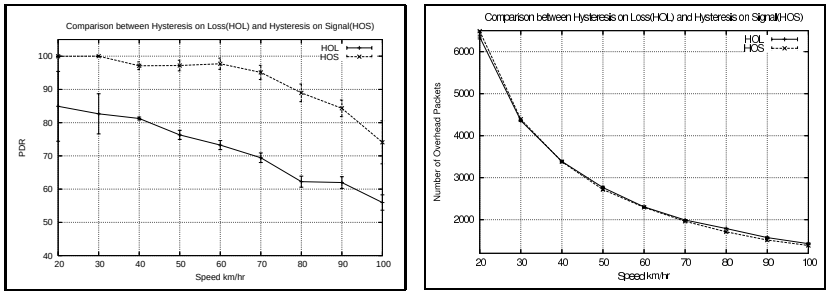
Sum_sig_var = 0
ENDIF
ENDIF
Change the status of Link if Link_quality crosses any of the two thresholds and reinitialize Sum_sig_var
ENDIF

Dans l'algorithme ci-dessus, *ss* est la valeur de la puissance du signal du dernier `Hello` reçu ; *Last_ss* est la puissance de l'avant dernier `Hello` reçu et *Hyst_ss_scaling* est un paramètre permettant de jouer sur la façon dont on augmente ou diminue la métrique de qualité du lien. Enfin Δ est le paramètre auquel est comparé les différences de puissance. Plus précisement, l'algorithme augmente ou diminue la qualité du lien si il y a une améloriation/détérioration supérieure à Δ de la puissance du signal. Cette amélioration/déterioration peut courir sur plusieurs `Hello` succéssifs.

2.1. *Résultats de simulations*

Pour nos simulations, nous utilisons une implémentation de OLSR du simulateur réseau NS version 2 [FAL 05]. Nous montrons ici uniquement les résultats pour une topologie en chaîne. Elle consiste en une série de 10 noeuds statiques placés à 100 mètres les uns des autres. Un noeud mobile est initialement placé à 10 mètres du premier noeud statique, et se met en mouvement de manière à passer devant toute la chaîne de noeuds. La vitesse du noeud mobile varie de 20km/h à 100km/h. Le noeud mobile transmet alors deux paquets de 512 octets toutes les secondes au premier noeud de la chaîne. Nous comparons le PDR (Packet Delivery Ratio) définit comme le nombre de paquets reçus divisé par le nombre de paquets émis, pour l'algorithme d'OLSR (basé sur l'hystérésis) et notre propre algorithme. Chaque point est la moyenne de 8 simulations. A ces moyennes nous associons un intervalle de confiance à 95% (une présentation plus exhaustives des résultats est donnée dans [ALI 07]).

Les résultats de simulations sont présentés sur les figures 1(a) et 1(b). Notre algorithme est clairement plus performant que celui d'OLSR (noté *Hysteresis on Loss* sur les figures). Comme expliqué plus haut, la raison est qu'un noeud statique doit attendre 4 secondes après que le noeud mobile soit sortie de sa portée radio pour que le lien soit dévalider. On voit que pour notre algorithme, il y a un PDR proche de 100% excepté pour les grandes vitesses. Ceci est dû au fait que notre algorithme a été dimensionné pour des vitesses moyennes (60km/h). La comparaison de l'overhead montre que notre algorithme n'utilise pas plus de paquets de contrôle que le OLSR du RFC. En effet, nous n'utilisons pas de nouveau message de contrôle et n'augmentons pas la fréquence des `Hello`.



(a) Comparaison du PDR (b) Comparaison de l'overhead

Figure 1 – Comparaisons des PDR et des overhead pour le protocole natif d'OLSR (*Hysteresis on Loss*) et notre algorithme (*Hysteresis on Signal*).

3. Conclusion

Les résultats de simulations ont montré que la prise en compte de la puissance du signal rendait la gestion des adjacences plus robuste et permettait d'anticiper les ruptures de liens, diminuant ainsi nettement les pertes de paquets. De plus, celui-ci ne produit pas de messages de contrôle supplémentaire. Les travaux en cours ont pour but de donner des règles de dimensionnement de notre algorithme. En effet, les performances sont étroitement liées aux différents paramètres de l'algorithme, à la distance entre les noeuds statiques et à la vitesse des noeuds mobiles. Une comparaison de notre algorithme et des autres protocoles de gestion des adjacences prenant en compte la puissance du signal est également planifié.

4. Bibliographie

[ALI 07] ALI H. M., NAIMI A. M., BUSSON A., VÈQUE V., « An efficient link management algorithm for high mobility mesh networks », *MobiWac '07 : Proceedings of the 5th ACM international workshop on Mobility management and wireless access*, New York, NY, USA, 2007, ACM, p. 42–49.

[Cla 03] CLAUSEN (ED) T., (ED) P. J., « Optimized Link State Routing protocol (OLSR) », October 2003, RFC 3626, Experimental.

[CRI 04] CRISOSTOMO S., SARGENTO S., BRANDAO P., PRIOR R., « Improving AODV with preemptive local route repair », *Wireless Ad-Hoc Networks, 2004 International Workshop on*, mai/juin 2004, p. 223–227.

[FAL 05] FALL K., EDITOR K. V., « The ns Manual », UC Berkeley and LBL and USC/ISI and Xerox PARC., http ://www.isi.edu/nsnam/ns/ns-documentation.html, January 2005.

[RAI] RAISINGHANI V. T., IYER S., « Cross-Layer Design Optimizations in Wireless Protocol Stacks ».

Modèlisation de protocoles MAC pour réseaux de capteurs à l'aide de chaînes de Markov

Fadila Khadar, David Simplot-Ryl

IRCICA/LIFL, Univ. Lille 1
CNRS UMR 8022, INRIA Lille Nord-Europe, France
Email: {fadila.khadar,david.simplot}@lifl.fr

RÉSUMÉ. *La conception d'une pile de communication pour une application pour réseaux de capteurs est fastidieuse. En effet, il faut déterminer les protocoles les plus aptes à convenir à cette application bien particulière. Il faut donc disposer d'un moyen pour les comparer en tenant compte des caractèristiques de l'application et de ses besoins. Nous proposons une méthode permettant d'évaluer les performances d'un protocole MAC dans un réseau donné et en fonction du trafic généré par l'application. Cette analyse s'appuie sur l'utilisation des chaînes de Markov à temps discret. Cette méthode offre une séparation claire entre le comportement du protocole et la topologie étudiée, permettant ensuite de les réutiliser pour former d'autres scénarios.*

ABSTRACT. *Protocol stack design for wireless sensor network is a tedious work. One must choose the best protocol for specific application and should have the ability to compare protocols to see which one best fit the characteristics of the application and its needs. We propose a method that evaluate MAC protocols performances which takes into account traffic generated by the application. This analysis is done using Discrete Time Markov Chains (DTMC). This method offers a clear separation between the behavior of the protocol and the studied topology. This way, the description of the protocol has to be written only once and then be reused with other applications.*

MOTS-CLÉS : *réseaux de capteurs, modèlisation de performances, protocole MAC, chaîne de Markov*

KEYWORDS: *wireless sensor network,performance modelisation,MAC protocol, markov chain*

1. Introduction

La conception d'une application pour réseaux de capteurs peut être laborieuse. A cela s'ajoute la difficulté de concevoir la pile de communication qui doit gérer les communications entre les nuds du réseau pour faire remonter l'information. En effet, de nombreux protocoles effectuant la même tâche sont disponibles, et chacun peut obtenir des résultats meilleurs que les autres dans une configuration donnée, mais également de pietres résultats dans une autre. Notre objectif principal est donc de proposer une méthode permettant, à partir d'une configuration (matériel utilisé, topologie du réseau etc), de déterminer les protocoles qui obtiendront les meilleurs résultats, notamment en consommation d'énergie. S'agissant des protocoles MAC, nous proposons une modèlisation à l'aide de chaîne de Markov. L'originalité de notre approche réside dans le fait que nous effectuons une séparation claire entre le protocole et la topologie qui permet d'utiliser la description d'un protocole avec des topologies différentes sans avoir à la réécrire.

Notre modèle repose donc sur une description du comportement du protocole au niveau d'un capteur et sur la specification de la topologie. Un comportement et une topologie forment un *scénario* à partir duquel nous obtenons une chaîne de Markov à temps discret. L'étude de celle-ci permet d'évaluer les performances du protocole.

2. État de l'art

Les chaines de Markov ont déjà utilisées pour étudier des systèmes concurrents tels que les réseaux de capteurs. On peut notamment citer les travaux de T. Razafindralambo et F. Valois ([RAZ 06]) qui utilisent PEPA (Performance Evaluation Process Algebra) pour modèliser le protocole 802.11. PEPA utilise des chaines de Markov à temps continu et permet de décrire le système à l'aide d'une algèbre de processus. Cependant, la modularité dont nous avons besoin n'est pas disponible. Pour un même protocole et des topologies différentes, il faut réecrire toute la modèlisation du canal radio. De plus l'utilisation de chaînes de Markov à temps continu ne permet pas de modéliser des évenements simultanés. Une analyse analytique du protocole IEEE 802.11 est également proposée dans [BIA 00]. Celle-ci donne des résultats très précis mais présente deux inconvénients majeurs pour l'utilisation que nous voulons en faire. Le premier est que cette analyse n'est pas réutilisable avec un autre protocole sans une étude poussée de ce protocole. Le deuxiéme inconvénient est qu'elle n'est valable que pour un réseau saturé, or nous voulons pouvoir étudier n'importe quel type de trafic.

3. Description du scénario

Description de la topologie La topologie du réseau à étudier est représentée par un graphe dont les sommets représentent les nuds du réseau et les arcs entre deux sommets indiquent que deux nuds peuvent communiquer.

Description du protocole Le comportement d'un protocole est représenté par un graphe dont les sommets indiquent l'activité du capteur à l'instant t. Par exemple, le sommet $compute$ montre que le capteur est en train de faire des calculs (ou tout autre activité) qui n'a pas de rapport avec la radio et le protocole MAC. Il existe un arc entre deux sommets i et j du graphe si il est possible au capteur de passer de l'activité i à l'activité j en un pas. La figure 3 montre un exemple de protocole MAC (qui correspond en fait à l'absence de protocole MAC) décrit sous forme d'automate. Les arcs entre les états i et j sont annotés par des couples de type $c\ :\ p$ qui indiquent que si la condition c est satisfaite, alors le nud a une probabilité p de passer de l'état i à l'état j. Par exemple, la condition C_2 qui régit le passage de l'état $compute$ à l'état $receive$ s'écrit :

$$C_2 : (exactly\ 1\ IN\ start_send)\ AND\ (exactly\ 0\ IN(end_send\ OR\ send))$$

Ainsi, un nud qui effectue des tâches de traitement peut passer dans l'état $receive$ si exactement un de ses voisins commence à envoyer des données et qu'aucun autre n'est déjà en train de transmettre (ce qui engendrerait une collision au niveau du nud). La probabilité p sur l'arc entre $compute$ et $start_send$ indique que le capteur génère un paquet avec une probabilité p. Dans cet exemple, la génération de paquet se fait seulement dans l'état $compute$. On peut utiliser un autre automate que l'on associe à celui ci pour gérer la génération de paquet. Ainsi, des paquets peuvent être générés dans n'importe quel état. Cet autre automate correspond au buffer du capteur.

Modélisation du canal radio Dans notre exemple, on suppose une couche physique idéale : tous les paquets émis d'un nud sont correctement reçus par ses voisins. Il serait cependant d'intégrer un mécanisme permettant d'utiliser une couche physique plus réaliste. Il faudrait pour cela modifier légèrement le modèle de topologie. Nous considérons également qu'un nud dont deux voisins (ou plus) émettent ne peut recevoir aucun des messages émis.

4. Génération de la chaîne de Markov

Une fois que l'on dispose du comportement du protocole et de la topologie, on définit l'état initial de la chaîne de Markov comme étant la concaténation des états initiaux des nuds. On détermine ensuite les états atteignables jusqu'à ce qu'ils soient tous énumérés. Dans notre exemple, et pour une topologie à deux nuds, l'état initial serait $(compute, compute)$. Les états atteignables en un pas sont alors $(compute, compute)$, $(compute, listen)$, $(listen, compute)$, et $(listen, listen)$. La chaine obtenue comporte alors 27 états.

On peut ensuite déterminer par exemple le nombre de paquets envoyés et correctement reçus par tous les voisins du nud émetteur en définissant un motif d'état. Dans ce cas précis le motif d'état serait de la forme :

$$M\ =\ atleast\ 1\ (instate\ end_send)\ suchthat\ atmost\ 0\ not_in\ receive$$

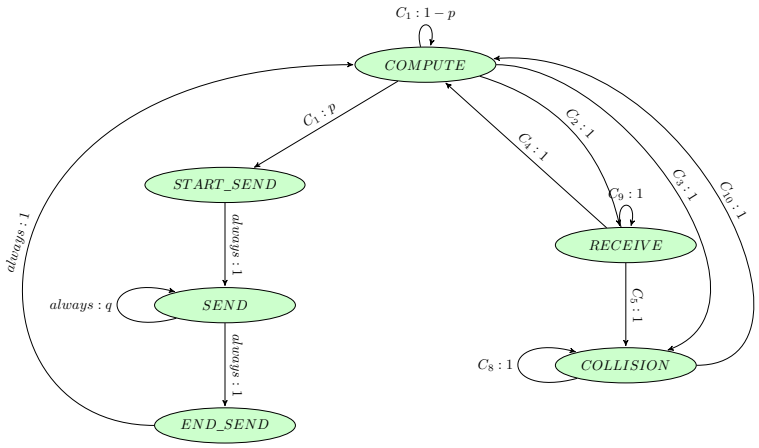


Figure 1. *Exemple d'automate*

Ce motif se lit : il existe, dans cet état, au moins un nud venant de terminer d'envoyer un paquet et dont tous les voisins sont dans l'état $receive$. Une fois la chaîne de Markov générée et les états auxquels on s'intéresse déterminés, la distribution stationnaire de la chaîne de Markov est calculé. Cette distribution permet de connaître la probabilité que l'on a d'être dans un état donné au temps t. *reformuler*

5. Conclusion et perspectives

Nous avons proposer une méthode de modèlisation des protocoles MAC offrant la possibilité de réutiliser la spécification d'un protocole pour étudier diverses topologies. Des améliorations à cette mèthode ont été apportées mais ne sont pas décrites dans cet article. Elles concernent notamment la mèthode de génération de paquets. Nous envisageons également d'utiliser l'algèbre tensorielle pour faciliter la résolution du système. Il sera ensuite possible de tester ce procédé avec des protocoles MAC plus élaborés. Il serait également intéressant d'intégrer une méthode simple pour changer le modèle de couche physique et le modèle d'interférences.

6. Bibliographie

[BIA 00] BIANCHI G., « Performance analysis of the IEEE 802.11 distributed coordination function », *Selected Areas in Communications, IEEE Journal on*, vol. 18, n° 3, Mar 2000, p. 535-547.

[RAZ 06] RAZAFINDRALAMBO T., VALOIS F., « Performance Evaluation of Backoff algorithms in 802.11 Ad-Hoc Networks », *3rd ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Malaga, Spain, octobre 2006.

Une approche modulaire probabiliste pour le routage à QdS intégrée.

Said Hoceini, Abdelhamid Mellouk, Hayet Hafi

LiSSi, IUT Créteil-Vitry, Université Paris XII
122 rue Paul Armangot Vitry sur Seine 94400 France
mellouk@univ-paris12.fr

RESUME. Les techniques traditionnelles de routage sont de plus en plus mal adapteés aux réseaux actuels. En effet, leur manque de réactivité vis à vis de la nature du trafic et au caractère variable des caractéristiques du réseau de transport les rendent souvent difficilement exploitables ou alors au prix d'un surdimensionnement des ressources du réseau (bande passante, mémoire tampon, charge CPU, etc.). L'objectif de cet article est de proposer un nouveau modèle algorithmique permettant de traiter la problématique du routage adaptatif dans un réseau de communication à trafic irrégulier et fortement dynamique. Celui-ci intègre des critères de QdS aussi bien dynamiques que statiques. Un exemple d'utilisation d'un tel modèle est donné ici : il repose sur l'apprentissage en continu des paramètres de routage, en particulier le temps bout-en-bout de remise des paquets l'état des files d'attente des routeurs. Les performances obtenues, comparativement aux approches classiques, sont très intéressantes dans le cas d'un trafic fortement dynamique.

ABSTRACT. Due to emerging real-time and multimedia applications, efficient routing of information packets in dynamically changing communication network requires that as the load levels, traffic patterns and topology of the network change, the routing policy also adapts. We focused in this paper on QoS based routing by developing a neuro-dynamic programming to construct dynamic state dependent routing policies. We propose an approach based on adaptive algorithm for packet routing using reinforcement learning which optimizes two criteria: cumulative cost path and end-to-end delay. Numerical results obtained with OPNET simulator for different packet interarrival times statistical distributions with different levels of traffic's load show that the proposed approach gives better results compared to standard optimal path routing algorithms.

MOTS-CLES. Routage multi chemins, Minimisation du délai de bout en bout, Routage avec QoS, Q-Routing, Apprentissage par renforcement.

KEYWORDS. Multi path Routing, Min. end-to-end delay, QoS Routing, Q-Routing, Reinforcement Learning.

1. Introduction

Le trafic dans les réseaux actuels se caractérise de plus en plus par des anomalies ayant pour conséquence des changements d'états imprévisibles dans le réseau [2] tout en nécessitant un transport garanti en terme de QdS. Il s'avère donc important de pouvoir s'adapter à la nature variable des paramètres d'un réseau ainsi qu'à la dynamicité des ressources disponibles. C'est ainsi que les approches développées ces dernières années ne se contentent pas seulement de manipuler des informations sur les plans de données, de contrôle et de gestion, elles intègrent de plus en plus des connaissances, acquises ou apprises, sur différents paramètres définissant l'état du réseau : trafic, ressources, besoins, etc.

L'objectif de cet article est de proposer un modèle de routage adaptatif capable de s'adapter aux conditions et à la nature d'un trafic dynamique d'une part et à l'utilisation des ressources du réseau d'autre part.

2. Le modèle proposé

Le problème posé ici est la mise en œuvre d'un algorithme de routage multi-chemins. Soit le graphe $G =(X, U)$ consistant en un ensemble X avec $|X| = N$ nœuds et un ensemble U avec $|U|= M$ liens. Les nœuds représentent les routeurs, commutateurs ou relais d'un réseau tandis que les liens représentent les tuyaux de communication (fibre optique, sans fil, etc.). Un lien spécifique de l'ensemble U entre les nœuds u et v est notée (u, v). Chaque lien $(u, v) \in U$ est caratérisé par un vecteur W de paramètres de dimension m, tel que $W(u,v)=[w_1(u,v), w_2(u,v),....., w_m(u,v)]$., où $w_m(u,v) \geq 0 \ \forall(u,v) \in U$ et les m composants se refèrent aux critèes de la QdS comme le délai, le coût, etc. Un chemin dans G est noté $P(s, t)$ s'il relie le nœud source s au nœud destinataire d. Les algorithmes de routage à QdS sont ceux qui calculent le chemin P, parmi les k chemins recherchés, optimisant une ou plusieurs contraintes liées à la QdS. Le vecteur L sera appelé vecteur de contraintes et les valeurs L_i mesurent les paramètres de QdS fixés par l'utilisateur (ou l'application). Les mesures de QdS sont de trois types : additif (ex. délai), multiplicatif (ex. taux de perte) ou min-max (ex. bande passante minimum nécessaire à un flux). On parle alors de typologie de métriques [3]. Au vu des multiples métriques w_i appelées à être optimisées, le problème du routage devient NP-complet dès qu'il s'agit d'optimiser deux critères (ou types de critères) non corrélés à la fois [4]. Le modèle que nous proposons, de complexité $O(kNlog(kN)+k^2mM)$, agit en deux étapes. Dans un premier temps, un ensemble de chemins potentiels est sélectionné au regard d'une fonction de coût à optimiser construite sur la base de critères statiques comme la bande passante, le nombre de sauts, le délai de bout en bout et le taux d'erreur (1).

$$\text{Coût}_{\text{statique}} = \text{f (bande passante, nombre de sauts, délai, taux d'erreur, etc.)} \quad (1)$$

Ensuite, le flux est distribué sur l'ensemble des chemins potentiels et le taux d'utilisation de chaque chemin est mis à jour en temps réel. Le coût de chaque chemin sélectionné dépend de l'évaluation des critères dynamiques tels que le taux d'acceptation, le taux de perte de paquet, le délai réel, la gigue, etc. Ceci exige également la définition d'une fonction de coût basée sur des critères dynamiques qui reste toujours un problème d'optimisation multi-critères (2).

$$Coût_{dynamique} = f'(\text{disponibilité, taux de perte de paquets, délai mesuré, gigue, bande passante mesurée, etc.}) \quad (2)$$

Nous allons détailler dans la suite une application du modèle proposé. Elle consiste dans le développement d'un algorithme basé sur un mécanisme d'apprentissage, issu du Q Learning, et permettant d'optimiser deux type de critères de QdS : un critère statique (le coût du lien) et un critère dynamique (le délai mesuré)

3. Application du modèle pour l'optimisation de deux critères : le coût et le délai de bout-en-bout.

Notre approche algorithmique est basée sur la technique du routage multi-chemins combiné avec le Q-Learning. L'espace d'exploration est réduit aux *K* meilleurs chemins au sens de critères statiques qui peuvent être liés par exemple à la bande passante, au coût des liens, au délai mesuré ou au taux de perte. Dans ce travail, nous nous sommes focalisés sur la recherche des chemins minimisant deux critères : le coût (pour des questions de simplicité, nous considérons des coûts unitaires) et le délai de bout en bout. Pour ce faire, nous utilisons l'algorithme de Dijkstra généralisé [5] afin de trouver les *K* meilleurs chemins sur la base des coûts des liens. La répartition du trafic se fera ensuite sur l'ensemble de ces *K* chemins en fonction du meilleur temps d'acheminement de bout en bout, celui-ci étant calculé par le biais d'un mécanisme basé sur l'apprentissage par renforcement. L'algorithme de mise à jour des paramètres de routage repose sur une méthode hybride associant le principe de l'exploration avancée à chaque fois qu'un paquet de données est échangé entre routeurs, à celui de l'exploration probabiliste permettant d'explorer les (*K -1*) autres chemins sans surcharger le réseau.

Pour résoudre le problème lié à l'exploration du réseau dans la phase d'apprentissage, une solution intéressante consiste à introduire un mécanisme permettant l'exploration régulière de tous les chemins. Pour notre approche, nous avons opté en premier lieu pour une exploration probabiliste (version appelée *KSPQR* de l'algorithme). Elle permet une exploration de temps à autre des (K-1) chemins pré sélectionnés par le premier module. Elle consiste à assigner arbitrairement une probabilité, que nous noterons P_{max}, au chemin optimal, les *(K-1)* autres chemins auront une probabilité équivalente à $(1 - P_{max})/K$. Une deuxième technique présentée dans cet article, appelée KOQRA, repose sur un calcul adaptatif de la probabilité de dis-

tribution prenant en compte les paramètres qui influent sur le choix du routeur : le délai estimé de bout en bout et le temps d'attente au niveau du routeur (temps de séjour dans la file d'attente). L'idée que nous proposons, issue de la théorie liée à l'intelligence collective des colonies de fourmis, consiste à introduire une procédure adaptative de calcul de probabilités pour chaque chemin, prenant en compte les deux paramètres définis précédemment. En adaptant la valeur de la probabilité, il s'agit de permettre au routeur d'éviter d'envoyer un paquet sur une interface dont le routeur d'extrémité possède une file d'attente saturée, même si le chemin dont ce dernier fait partie, présente le plus court délai.

4. Simulations

Les simulations, effectuées sur la plate forme OPNET, ont porté sur l'architecture japonaise *NTTnet* présenté dans [6] (à droite de la figure 1). Pour un but de comparaison, les performances des deux versions de l'algorithme (KSPQR et KOQRA) ont été comparées à ceux des algorithmes traditionnels SPF (*Shortest Path First*) et SOMR (*Standard Optimal Multi-Path Routing*) en termes de temps d'acheminement moyen des paquets.

L'analyse des caractéristiques s'est faite au niveau de l'entité la plus fine qu'est le paquet en utilisant un modèle poissonnien pour simuler le trafic. L'ensemble de nos évaluations sont faites en dehors de tout contrôle supplémentaire se faisant au-delà de la couche réseau comme par exemple un contrôle d'erreurs ou un ordonnacement de paquets par la couche transport. Des travaux futurs doivent nécessairement les prendre en compte dans le cas d'un déploiement à grande échelle et d'une étude portant sur leur interaction avec l'ensemble des composants d'un réseau.

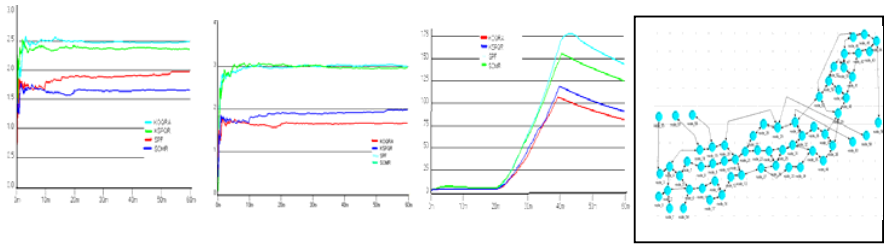


Figure 1 : Temps moyen d'acheminement pour un trafic (gauche : trafic faible, milieu : trafic fort, droite : pic de trafic) et le réseau NTT.

Sur l'ensemble des figures représentant les résultats des différentes expériences (Figure 1), l'axe des abcisses représente le temps de simulation tandis qu'une indication sur le délai moyen d'acheminement des paquets est représentée au niveau de l'axe des ordonnées. Les résultats de simulation obtenus sur le réseau *NTT*, dans des conditions de trafic faible, et représentés sur la partie gauche de la figure 1, montrent que les temps moyens d'acheminement sont plus favorables aux algorithmes traditionnels SPF et SOMR comparativement aux autres algorithmes basés sur les

techniques adaptatives, où on constate que le temps moyen d'acheminement est augmenté de près de 15%. Comme ces dernières sont basées sur un apprentissage continu par le biais des paquets de contrôle qu'elles génèrent, ces derniers surchargent inutilement le réseau et font chuter par conséquent les performances du routage. Dans le cas où le trafic sur le réseau devient très chargé, les résultats illustrés dans la partie centrale de la figure 1 montrent distinctement que les approches adaptatives donnent de meilleures performances en termes de temps moyen d'acheminement des paquets comparativement aux algorithmes classiques (SOMR et SPF). Comme le facteur lié à la congestion est pris en compte d'une façon adaptative, dynamique et qualitative dans les algorithmes KOQRA et KSPQR, ces derniers établissent de nouvelles routes en fonction de leurs qualités (temps d'attente et temps de transit) à chaque fois qu'une route en cours d'utilisation s'avère encombrée. Les algorithmes classiques (SOMR et SPF) mettent quant à eux plus de temps à réagir à cette congestion au regard de la périodicité utilisée dans les mises à jour de leurs paramètres et à leur estimation des paramètres de routage de manière locale. Le même résultat est constaté dans le dernier scénario testé (courbe de droite de la figure 1) où les conditions d'un pic de trafic sont créées.

6. Conclusion

Dans cet article, nous avons présenté une nouvelle approche basée sur la technique du routage multi-chemins combiné avec des algorithmes de type adaptatif. Une analyse comparative des performances de l'approche proposée montre clairement l'efficacité du modèle proposé et son intérêt pour des réseaux à forte charge ou soumis à des changements soudains conduisant à des pics sporadiques de trafic. Comme l'optimisation d'une fonction de coût composée de plusieurs critères non corrélés est un problème NP-complet, nous nous proposons comme suite de ce travail d'étudier, sur la base des algorithmes que nous avons développés, les possibilités d'intégrer à la fois plusieurs paramètres de QoS (bande passante et taux de perte des paquets) dans le signal de renforcement.

7. Bibliographie

[1] A. Mellouk, P. Lorenz, A. Boukerche, M. H. Lee, «*Quality of Service based Routing Algorithms for Heterogeneous Networks »*, numéro spécial de la revue « IEEE Communication Magazine », Vol. 45, n°2, pp. 65-112, Février 2007.

[2] N. Larrieu, « *Proposition d'une architecture de gestion du trafic à partir de mesures reposant sur un mécanisme de contrôle de congestion permettant l'optimisation de la QdS dans l'Internet* », thèse de doctorat de l'INSA de Toulouse, 2005.

[3] Z. Wang and J. Crowcroft, "*QoS Routing for Supporting Resource Reservation*". In IEEE Journal on Selected Areas in Communications, Septembre 1996.

[4] J. Li, T. McCormick et D. Smitchi-Levi, '*The Complexity of Finding Two Disjoint Paths with Min-Max Objective Function*', In Discrete Applied Mathematics, 26:105-115, 1990.

[5] D Eppstein, "*Finding the K shortest paths*", SIAM J. Computing 28:0, 1999, pp. 652-673.

[6] G. Di Caro, M. Dorigo, "*Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks*", Free Brussels University, Ed. MIT Press, Septembre 2004.

Modèle générique pour le routage orienté contenu

Yosra Barouni* — **Prométhée Spathis*** — **Serge Fdida***

** Laboratoire d'informatique de Paris 6*
CNRS/Université Pierre et Marie Curie
104 avenue du président kennedy, 75016 PARIS
{yosra.barouni, promethee.spathis, serge.fdida}@lip6.fr

RÉSUMÉ. *Nous proposons un modèle générique pour capturer les propriétés distinctives des diverses approches existantes en matière de distribution de contenu. A partir de ce modèle, nous introduisons une terminologie permettant d'analyser les fonctionnalités propres aux approches existantes. Ensuite, nous proposons une architecture de routage orientée contenu effectuant un filtrage atomique des requêtes basé sur un vecteur optimal calculé pour un ensemble de métriques de filtrage.*

ABSTRACT. *In this paper, we propose a generic model that captures the inherent properties of the existing approaches that resolve the content routing and distribution field. From this model, we propose a terminology that allows us to analyse the functionalities implemented in these solutions. Then, we propose a content based routing architecture that filters client request using an optimized filtering vector.*

MOTS-CLÉS : *Réseaux de contenu, routage, filtrage, optimisation.*

KEYWORDS: *Content based networks, routing, filtering, optimization*

1. Introduction

Le succès que connaît l'Internet est largement dû la quantité de contenu mis è la disposition de tous. Pourtant, l'infrastructure aussi bien matérielles que logicielles de l'Internet ainsi que son modèle orienté hôte n'ont pas été architecturés pour permettre la livraison de contenu à son échelle. Pour être rendus, les services en rigueur requièrent qu'une connexion entre adresses IP soit préalablement établie. L'intérêt grandissant dans les architectures orientées contenu comme les systèmes pair-à-pair, les réseaux de distribution de contenu ou les systèmes basés sur les publications / abonnements se situe au niveau de la proposition d'un modèle de routage centré sur les contenus plutôt que les hôtes. Chacune des ces approches ayant été proposées pour répondre aux besoins spécifiques de certaines applications, il n'existe pas de solution générique qui s'accommode des réseaux et d'éléments ainsi que la complexité d'interaction entre ces approches différentes. Il est donc nécessaire de développer des solutions architecturales globales permettant le routage et la distribution du contenu et pouvant interconnecter les différentes infrastructures de gestion de contenu et distribuer le contenu à travers elles.

Dans cet article, nous présentons un modèle générique qui capture les propriétés distinctives des diverses approches proposées en matière de distribution de contenu. Notre modèle introduit une terminologie à partir de laquelle nous analysons les fonctionnalités propres aux approches existantes. A partir de cette analyse, nous proposons une architecture de routage orientée contenu basée sur le filtrage atomique des requêtes. L'acheminement des requêtes est réalisé de proche en proche en se basant sur un vecteur optimal calculé pour un ensemble de métriques de filtrage.

2. Travaux similaires

Dans la littérature, plusieurs solutions typiques aux routage orienté contenus ont été proposées comme les réseaux de distribution de contenus (CDN) ou les systèmes basés sur les événements dont ceux basés sur les publications/abonnements.

Réseaux de distribution de contenu Au niveau des réseaux de distribution de contenu (CDNs) [Hul 02], le routage consiste à distribuer le contenu à partir du meilleur emplacement vers le client qui le demande. Le processus de filtrage et de décision revient donc à choisir le meilleur emplacement de distribution du contenu plutôt que de filtrer le contenu adéquat aux intérêts des clients [DCT 03]. Plusieurs métriques peuvent affecter cette décision comme la proximité en terme de sauts entre les clients et les serveurs de contenu ou la proximité géographique. Les CDNs constituent une solution qui optimise la distribution des contenus en terme de rapidité et de disponibilité mais ne sont toujours pas centrés sur les propriétés des contenus dans le processus de routage et ne peuvent être déployés tels qu'ils sont conçus actuellement à l'échelle d'Internet.

Systèmes basés sur les événements Au sein des systèmes basés sur les publications / souscriptions [EFG 03], les clients notifient leur abonnement à un type de contenu à l'infrastructure de gestion d'événements. Cette infrastructure se charge de faire correspondre les abonnements reçus par rapport aux publications qu'elle stocke. Ensuite, les

clients sont notifiés chaque fois qu'un contenu correspondant à leurs critères d'abonnement est trouvé. Les avantages d'une telle architecture sont la persistance des abonnements ainsi que le découplage entre les requêtes clientes et les réponses reçues. Mais elle n'offre pas pas de filtrage optimisé et pas de tri des réponses les plus optimales envoyées au client.

Pour résoudre la problématique du routage centré exclusivement sur le contenu et non sur les hôtes, nous proposons de router les contenus uniquement à la base de leurs descriptions tout en optimisant un ensemble de métriques de filtrage. Notre solution est différente car elle imbrique les mots clés décrivant le contenu dans le processus du routage lui-même. Le routage réel et le filtrage du contenu sont combinés en un même processus permettant ainsi un gain conséquent en terme de messages échangés sur le réseau. Elle diffère par sa récursivité et a structure hiérarchique qui permettent de prendre en considération les changements de topologies et d'informations sur les contenus.

3. Modèle générique pour la routage orienté contenu

Les réseaux orientés contenu font référence à la capacité donnée aux applications et à leurs utilisateurs d'accéder aux données indépendamment de leur localisation. Les contenus sont accédés sur la seule base de leur type ou de leur description . Ces réseaux s'affranchissent de la nécessité d'identifer dans un système de nommage ou d'adressage les hôtes hébergeant les contenus ou souhaitant les accéder.

Entités impliquées Les réseaux orientés contenu sont constitués de noeuds qui diffèrent selon le rôle qui leur est attribué. On distingue les fournisseurs d'origine, les clients, les routeurs de médiation et les serveurs de contenu.

Processus d'accès au contenu Les fournisseurs d'origine disséminent les informations décrivant suivant une structure donnée les contenus qu'ils souhaitent rendre accessibles. Pour être en mesure de recevoir des contenus pertinents, un client définit et structure ses centres d'intérêt avant de les annoncer. Entre ces deux acteurs, les serveurs de contenu et les routeurs de médiation jouent des rôles primordiaux. Les serveurs de contenu gèrent la réplication et la mise en cache des données les plus accédées dans le but de minimiser leur temps d'accès et de diminuer la charge de traitement nécessaire aux fournisseurs d'origine pour satisfaire les requêtes de client. Les routeurs de médiation quant à eux gèrent la dissémination vers leurs pairs des requêtes de client et des annonces de fournisseurs. Ils stockent aussi des tables de correspondance entre les états concernant une partie ou la totalité des contenus disponibles et la localisation des fournisseurs d'origine hébergeant ou répliquant ces contenus. Ces tables permettent aux routeurs de médiation d'exécuter des algorithmes de filtrage de contenu et de redirection des requêtes de client de manière centralisée ou distribuée dans le but de sélectionner le(s) contenu(s) correspondant aux intérêts annoncés des clients. Notons que les quatre rôles que nous renons de définir peuvent être assumés par un même et unique nœud comme dans les réseaux de capteurs. Ce processus d'accs au contenu peut être déployé selon les contextes d'application à travers modèles différents : le modèle applicatif travers un réseau virtuel comme les systèmes pair à pair, le modèle réseau travers un filtrage XML intégré dans les routeurs niveau 3 et le modèle hybride combinant les deux comme dans le cas de certains réseaux CDNs.

4. Routage orienté contenu : algorithme et optimisation

L'idée sous-jacente à notre proposition est de substituer le routage classique basé sur les adresses des destinations par un routage saut-par-saut du meilleur contenu décrit avec un ensemble de termes $\{K_1...K_n\}$. La solution que nous proposons passe le facteur d'échelle en organisant les routeurs de médiation dans une structure hiérarchique en domaines. La dissémination des informations décrivant les contenus se limite au voisinage intra-domaine du routeur de médiation.

D'après le modèle général proposé dans la section 3, le problème de routage centré sur le contenu peut donc être réduit aux points suivants :

– Quelles sont les métriques à partir desquelles un routeur de médiation filtre les requêtes ?

– Dans quel type de structure ces métriques sont-elles stockées et comment sont-elles calculés ?

– Comment se fait le filtrage des contenus au niveau du routeur médiateur lui-même ?

– Quelles sont les politiques de routage et de filtrage inter et intra-domaines ?

Les types de requêtes pouvant varier d'une application cliente à une autre, nous commençons par transformer les requêtes reçues par l'infrastructure de routage orienté contenu en un format générique composé d'un ensemble de mots-clés $\{K_1...K_n\}$. Chaque mot-clé K_i décrit les propriétés susceptibles d'être intéressantes dans un contenu. Une relation de correspondance entre cette description générique et le format initial est stockée grâce dans une structure dédiée au sein de routeurs de médiation spécifiques. Ainsi, nous garantissons la compatibilité des requêtes de client indépendamment des applications à l'origine de la recherche initiée.

Une des contributions de notre solution est de baser le filtrage sur le calcul d'un vecteur de filtrage noté $[D, P, M]$ dont chaque coordonnée correspond à une métrique de filtrage de contenu. D représente la distance entre le client et les potentiels contenus correspondant à sa requête en nombre de sauts. P représente la popularité du contenu en nombre d'accès total pour des requêtes similaires. Quant à M, elle représente le taux de requêtes satisfaites par le système. Le calcul de la popularité d'un contenu utilise les statistiques d'accès au contenu réalisées à partir de l'historique. Nous nous basons aussi sur un système de gestion de la *réputation* des contenus pour calculer les valeurs de la troisième métrique. Les trois métriques que nous venons de présenter sont stockées au niveau de la table de filtrage. Les métriques sont reliées à l'identifiant local d'une interface de sortie ce qui permet de diriger la requête générique préalablement créée dans la direction indiquée par le vecteur $[D, P, M]$ optimal associé tel que représenté la figure 1.

Dans cette figure, une requête est reçue sur une des interfaces entrantes du routeur de médiation. Un module de filtrage retire les mots-clés contenus dans cette requête pour les insérer dans deux sous-requêtes atomiques. Chacune d'elles est traitée séparément et le mot-clé qu'elle contient est comparé aux valeurs contenues dans la colonne des mots-clés $Dest_Key$. Pour chaque sous-requête atomique, le module retourne l'in-

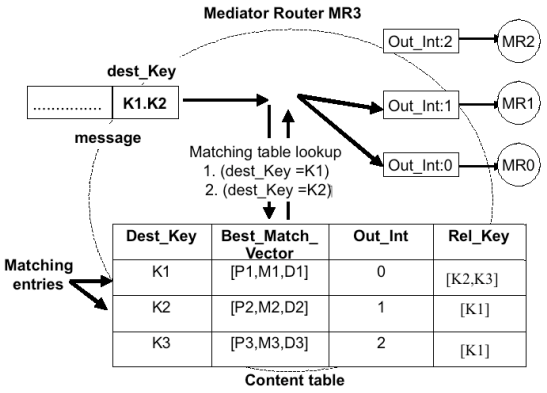


Figure 1. *Routage et filtrage du contenu à l'intérieur d'un routeur de médiation*

terface sortante correspondant au champ Ou_Int. Les résultats des sous-requêtes sont agrégés afin d'éliminer les interfaces redondantes et les doublons. Ensuite, la requête est redirigée alors dans sa totalité vers les différentes interfaces sortantes résultantes. Notons que le filtrage tel que présenté dans la figure 1 est réalisé de proche-en-proche. Cette propriété permet de prendre en considération les changements de topologie dûs à la mobilité des noeuds ou aux défaillances du réseau ainsi que la latence de propagation des informations de routage dans le réseau.

5. Conclusion et perspectives

Nous avons présenté dans ce document un modèle de réseau qui capture les propriétés des réseaux orientés contenu qui nous as permis de les étudier dans des contextes différents. En nous basant sur les fonctionnalités et la terminologie qu'introduit notre modèle, nous avons décrit une proposition de filtrage de contenu. Notre solution présente la particularité d'être résistante au facteur d'échelle et de s'accommoder de l'hétérogénéité des protocoles existants. Nous prévoyons de définir les comportements exacts nécessaires à l'implémentation de notre proposition.

Bibliographie

[DCT 03] DAY M., CAIN B., TOMLINSON G., RZEWSKI P., *RFC3466, A Model for Content Internetworking (CDI)*, RFC Editor, 2003.

[EFG 03] EUGSTER P., FELBER P.A., GUERRAOUI R., KERMARREC A.M., *The many faces of publish/subscribe*, *ACM Comput. Surv.*, vol. 35, n° 2, 2003, p. 114-131.

[Hul 02] HULL S., *Content Delivery Networks : Web Switching for Security, Availability, and Speed*, McGraw-Hill Inc, 2002 .

[PGM 06] PLAGEMANN T., GOEBEL V., MAUTHE A., MATHY L., TURLETTI T., URVOY-KELLER G., *From content distribution networks to content networks - issues and challenges*, *Computer Communications (COMCOM)*, vol. 29, n° 5, 2006, p. 551-562.

Effet de la mobilité MANET sur un système P2P

S. Ktari[1,2], A. Hecker[2], F. Springinsfeld[1]

[1] *Wavestrom SAS R&D, 40 rue d'Oradour sur Glane, 75015 Paris*
{sktari,fs}@wave-storm.com
[2] *TELECOM ParisTech & LTCI (CNRS UMR 5141), 46 rue Barrault, 75013 Paris*
{hecker}@enst.fr

RÉSUMÉ. *Dans ce travail, nous analysons l'impact de la mobilité des nœuds, gérées par un algorithme de routage MANET, sur un système pair à pair de localisation de ressources structuré. Dans ce contexte, nous évaluerons, par le moyen d'émulations, les performances de la DHT de Chord dans un réseau MANET employant AODV et OLSR.*

MOTS-CLÉS : *MANET, réseau P2P structuré, mobilité, émulation.*

1. Introduction

Certains points de convergence entre les réseaux MANETs et P2P tels que la décentralisation, l'auto organisation et la spontanéité motivent le croisement des deux domaines. D'autres en revanche, laissent penser qu'un tel choix n'est pas fondé. En effet, la non fiabilité des liens, la volatilité des nœuds et la charge importante de signalisation met en doute l'efficacité de tels systèmes. Cette problématique a été déjà traitée et plusieurs solutions ont été proposées (Thomas *et al* 2006) (Thomas *et al.* 2005) (Charlie *et al.* 2003). Les approches présentées proposent des systèmes complets intégrant les deux domaines. Toutefois, l'évaluation de tels systèmes restent le plus souvent basée sur les techniques de simulation ou, rarement, sur des expérimentations réelles dans des réseaux de taille très réduite.

Dans ce travail, nous évaluons, à l'aide d'émulations, l'impact de la volatilité des nœuds, gérées par un algorithme de routage MANET, sur un système P2P de localisation des ressources structuré. Nous identifions notamment les différents taux de mobilité au-delà desquels les performances du système tombent au dessous d'un seuil requis.

2. Evaluation des performances de la DHT de Chord dans un MANET

L'objectif de ce travail est d'évaluer la faisabilité de déploiement d'un réseau P2P MANET, dans un environnement quasi réel moyennant des émulations. Nous évaluerons, faisant varier le paramètre *churn* (nombre de connexion/déconnexion), les paramètres suivants :

- Le taux de succès des recherches : le pourcentage des requêtes réussies. Dans notre modèle, une requête qui n'aboutie pas au bout de 2 essaies successifs est considérée comme échouée.
- Le délai de recherche : le délai moyen des requêtes abouties.
- Le nombre de message de signalisation (paquet) : le nombre de messages de contrôle générés par la DHT et le protocole de routage ad hoc.
- La quantité du trafic de signalisation (en Mo) : la taille totale des messages de contrôle en Méga octet généré pour la maintenance de la DHT et pour la maintenance du routage ad hoc.

Dans ce qui suit, nous décrivons notre modèle d'émulation, les protocoles étudiés et les résultats émanant.

2.1. *Plateforme des émulations*

Dans notre projet, nous avons choisi d'utiliser User Mode Linux (UML) avec Netkit (Massimo R 2007) comme plateforme principale. Le but et de permettre la mise en place d'un réseau complètement virtuel de machines Linux, pouvant communiquer entre elles, sur une ou quelques machines physiques. Pour le routage physique ad hoc, nous avons choisi le tester un protocole réactif (AODV UU v9.5) et un protocole proactif (OLSR v5.0). Sur le niveau applicatif d'un nœud, nous avons choisi la DHT de Chord (réalisée en C).

Nous émulons un réseau de 80 nœuds partageant 1000 clés. L'arrivée et le départ des nœuds (*churn*) suivent une loi de Poisson de moyenne variable (entre 1 churn /5 min et 0 : réseau statique). Le *churn* peut simuler dans une certaine mesure la mobilité des nœuds puisqu'il représente également des connexions (nouveau voisin) et des déconnexions (départ d'un voisin) comme dans les réseaux mobiles. A intervalles réguliers, 20 nœuds génèrent en parallèle une requête de recherche vers une clé choisie aléatoirement. Le timeout et le nombre d'essaie pour chaque requête sont fixés respectivement à 5 sec et 2.

2.2. *Résultats*

La figure 1 présente les résultats de performances des systèmes OLSR_Chord et AODV_Chord, faisant varier le paramètre *churn*.

Dans un réseau statique, on récupère 100% des clés pour les deux systèmes. Plus la mobilité augmente, moins les requêtes aboutissent. Le protocole AODV_Chord est plus sensible aux *churns.* En effet, pour toutes les valeurs de *churn,* le taux de

succès est inférieur à 95% et le délai est nettement supérieur à celui d'OLSR_Chord. Le protocole OLSR_Chord produit un taux de succès qui reste supérieur à 95% jusqu'à un *churn* seuil égal à 1/15. De plus, OLSR_Chord produit un délai nettement meilleur surtout pour les valeurs de *churn* élevées. La maintenance proactive d'OLSR réduit l'effet du *churn* puisque les routes sont vérifiées périodiquement et donc réparées immédiatement en cas de pannes.

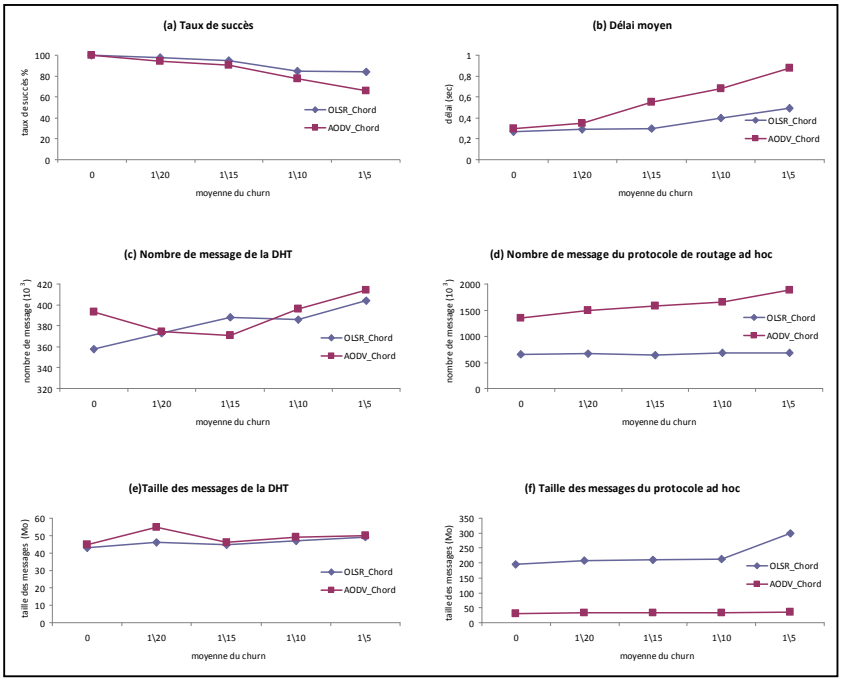


Figure 1. *Effet du churn*

Nous avons choisi de calculer la quantité de signalisation produite par la DHT et le protocole de routage séparément. Nous remarquons que la signalisation générée par le protocole de routage est nettement supérieure à celle de la DHT. Aussi, le nombre et la taille des messages générés pour la maintenance de la DHT varient très peu avec le *churn*. Par ailleurs, le nombre et la taille des messages générés par le protocole de routage diffèrent. AODV_Chord, bien qu'il soit réactif, génère plus de messages de signalisation. Toutefois, la taille totale des messages d'OLSR_Chord est nettement supérieure. La diffusion par MPR dans OLSR réduit le nombre de messages véhiculés, ce qui explique le premier résultat. Or, dans OLSR les nœuds s'échangent périodiquement leurs tables de routage et de voisinage. Ces messages décrivent toute l'architecture du réseau, et donc leur taille dépend de la taille du réseau ; d'où la taille importante du trafic de signalisation. Dans un réseau de 80

nœuds, la taille moyenne d'un message OLSR est de 331 octets contre 20 octets pour AODV. Nous avons aussi calculé ces valeurs pour un réseau de 40 nœuds avec les mêmes paramètres. La taille moyenne d'un message AODV est égale à 23 octets et celle d'un message OLSR est de 170 octets.

L'utilisation du protocole OLSR_Chord est donc limitée par la taille du réseau. La signalisation induite croît fortement avec le nombre de nœuds et inhibe les performances du système. AODV_Chord, en revanche, génère une signalisation moins lourde mais produit des performances moindre, notamment avec un *churn* élevé. Les facteurs qui influent le plus les performances des systèmes P2P MANET résultant sont donc la mobilité pour AODV_Chord et la taille du réseau pour OLSR_Chord.

3. Conclusion

Dans ce travail, nous avons présenté une analyse quantitative évaluant l'impact de la volatilité des nœuds MANET, sur un système pair à pair structuré. Les résultats montrent que le protocole OLSR_Chord présente des performances meilleures que celles d'AODV_Chord, mais produit aussi une forte signalisation et consomme plus de ressources. OLSR_Chord conviendrait alors dans le cas des réseaux de tailles moyennes exigeant un délai minimal et une haute fiabilité. Les résultats montrent aussi que le protocole AODV_Chord est peu approprié aux réseaux hautement dynamiques, et par conséquent aux systèmes P2P.

4. Bibliographie

Thomas Fuhrmann, 'Combining Virtual and Physical Structures for Self-organized Routing', *IWSOS/EuroNGI 2006.*

Thomas Zahn and Jochen Schiller, 'MADPastry: A DHT Substrate for Practicably Sized MANETs', In Proc. of *ASWN*. June 2005

Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha, 'Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks', In Proc. *HotOS IX Workshop*, pages 37--42, Lihue, HI, USA, 2003

Massimo Rimondini, 'Emulation of Computer Networks with Netkit', rapport RT-DIA-113-2007 January 2007.

Ion Stoica , Robert Morrisz, David Liben-Nowellz, David R. Kargerz, M. Frans Kaashoekz, Frank Dabekz,Hari Balakrishnanz, 'Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications', *ACM SIGCOMM, , San Diego, California*, August 2001.

Index des auteurs

Index

Berthou, Pascal, 30

Aïache, Hervé, 131
Aiache, Hervé, 55
Aib, Issam, 339
Ali Kaafar, Mohamed, 266
Ali, Husnain Mansoor, 362
Anelli, Pascal, 239
Armando, François, 251
Augustin, Brice, 290

Barouni, Yosra, 376
Barrenetxea, Guillermo, 144
Begin, Thomas, 5
Ben-El-Kezadri, Ryad, 339
Berthou, Pascal, 94
Brandwajn, Alexandre, 5
Buob, Marc-Olivier, 278
Busson, Anthony, 357, 362

Cantin, François, 266
Carle, Jean, 156
Cavalli, Ana, 81
Chassot, Christophe, 251
Cherif, Mohamed Oussama, 327
Conan, Vania, 131
Czachórski, Tadeusz, 18

Ducourthial, Bertrand, 327

Exposito, Ernesto, 251

Farraposo, Silvia, 69
Fdida, Serge, 107, 376
Festor, Olivier, 214
Friedman, Timur, 290

Gallais, Antoine, 156
Gauchard, David, 94
Gayraud, Thierry, 30
Gelas, Jean-Patrick, 42
Grad, Dominique, 227
Grochla, Krzysztof, 18
Guerin Lassous, Isabelle, 315
Gueye, Bamba, 266

Hafi, Hayet, 371
Harivelo, Fanilo, 239
Hecker, Artur, 381
Heurtefeux, Karel, 177
Hilt, Benoît, 227
Hoceini, Said, 371

Iannone, Luigi, 107
Ingelrest, François, 144

Jacquemin, Baptiste, 30
Jerbi, Moez, 327

Kaisser, Florent, 357
Khadar, Fadila, 367
Khalfallah, Sofiane, 327
Ktari, Salma, 381

Labiod, Houda, 381
Labit, Yann, 94
Lauriano, Mattéo, 55
Layouni, Farah, 290
Lebrun, Laure, 131
Leduc, Guy, 266
Lee, Joongsoo, 168
Leguay, Jérémie, 131
Lochin, Emmanuel, 239
Lucas, Vincent, 227

Martins, Eliane, 81
Mathy, Laurent, 266
Mellouk, Abdelhamid, 371
Meulle, Mickael, 278
Mohamed, Bezahaf Mehdi, 107
Molle, Christelle, 119
Monteiro, Edmundo, 69
Morais, Anderson, 81
Moreira, Bruno, 81
Mornard, Olivier, 42

Naimi, Amina Meraihi, 362
Nussbaum, Lucas, 352

Owezarski, Philippe, 69, 94

Pansiot, Jean-Jacques, 227
Peix, Fabrice, 119
Pekergin, Ferhan, 18
Perros, Kevin, 168
Popi, Cristian, 214
Primet, Pascale, 42
Pujolle, Guy, 339

Richard, Olivier, 352
Rivano, Hervé, 119
Rousseau, Stéphane, 131

Sawma, Gilbert, 339
Senouci, Sidi-Mohammed, 327
Sieux, Corinne, 55

Silverston, Thomas, 202
Simplot-Ryl, David, 367
Spathis, Prométhée, 376

Tavernier, Cedric, 55
Teixeira, Renata, 290
Thoumin, Damien, 131
Toutain, Laurent, 168
Tredan, Gilles, 190

Uhlig, Steve, 278

Vèque, Véronique, 357, 362
Val, Thierry, 303
Valois, Fabrice, 177
Van den Bossche, Adrien, 303
Van Wambeke, Nicolas, 251
Vannier, Rémi, 315
Vetterli, Martin, 144
Viana, Aline, 190