

Sélection Dynamique de la Méthode de Compression

Rayene Ben Rayana, Jean-Marie Bonnin, Abdelfattah Belghith

► **To cite this version:**

Rayene Ben Rayana, Jean-Marie Bonnin, Abdelfattah Belghith. Sélection Dynamique de la Méthode de Compression. Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006, Oct 2006, Tozeur/Tunisia, Hermès, Session 1 : compression

robustesse, 12 p., 2006. <inria-00110249>

HAL Id: inria-00110249

<https://hal.inria.fr/inria-00110249>

Submitted on 20 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sélection Dynamique de la Méthode de Compression

Rayene BEN RAYANA* — Jean-Marie BONNIN** — Abdelfattah BELGHITH***

*** ENST-Bretagne, 2 Rue de la châtaigneraie, CS 17607
35576, Cesson-Sévigné - France.

* rayene.benrayana@enst-bretagne.fr, ** jm.bonnin@enst-bretagne.fr

*** ENSI, Campus universitaire de la Manouba
2010, La Manouba - Tunisie
abdefattah.belghith@ensi.rnu.tn

RÉSUMÉ. Les méthodes de compression réagissent différemment selon les conditions réseau et le type de flux à compresser. De ce fait, le choix d'une méthode de compression est associé à l'usage d'un type de flux particulier soumis à des conditions réseau particulières. Le non respect de cette association conduit généralement à une dégradation des performances de la compression. Il serait donc intéressant de sélectionner dynamiquement la méthode de compression la plus adaptée à chaque flux en fonction du contexte d'exécution. Pour renseigner un tel algorithme de sélection, il est nécessaire de définir un mécanisme capable de prévoir les performances de chaque méthode de compression pour un flux donné. Ce mécanisme nécessite une phase d'étalonnage préliminaire réalisée grâce à une campagne de mesures. Cet article décrit les problèmes résultant de l'utilisation d'une unique méthode de compression, introduit les bases de l'algorithme de sélection et présente les résultats de la campagne de mesures.

ABSTRACT. Compression protocols do not behave the same manner under different network conditions and for different flow types. The choice of a compression method constrains the use of particular flow types under particular link characteristics. If not respected, the performances of compression could be degraded. A suitable solution would be a dynamic selection of the best compression method given a flow and link characteristics. To guide such a selection algorithm, we had to define a mechanism able to predict the performances of each compression method on a given flow. This mechanism needs a preliminary calibration achieved by a measurement campaign. This paper addresses issues arising from the use of a single compression protocol, describes the results of the measurement campaign and introduces the basics of the automatic selection protocol.

MOTS-CLÉS : Compression, PPP, IPv6, mobilité, tunnel

KEYWORDS: Compression, PPP, IPv6, mobility, tunnel

1. Introduction

Avec l'expansion des réseaux cellulaires et des réseaux locaux sans-fil et les travaux autour de la mobilité dans ces réseaux, les méthodes de compression actuelles sont devenues obsolètes. En effet, ces méthodes assument une certaine stabilité des conditions réseau et ne sont généralement adaptées qu'à un seul type de flux. Aujourd'hui, le support de la mobilité tend à se généraliser et, dans un contexte de mobilité, les conditions réseaux peuvent difficilement être stabilisées. De plus, les terminaux mobiles sont de plus en plus polyvalents. Ils sont utilisés en tant que téléphones, visiophones, clients de messagerie instantanée, lecteurs multimédia, etc. Ainsi, les liens sans-fil sont utilisés pour véhiculer des flux hétérogènes.

Nous proposons une architecture intégrant un algorithme de sélection dynamique de la méthode de compression la plus adaptée au contexte. Ce dernier sépare le trafic en flux et essaie de prédire, à partir d'une évaluation des conditions réseau et des caractéristiques du flux, les performances que vont donner chacune des méthodes de compression. Il peut ainsi sélectionner, pour chaque flux, la meilleure manière de le compresser.

Afin d'effectuer ces prédictions, cet algorithme requiert un étalonnage préalable nécessitant une campagne de mesures. Ces mesures, tentent d'imposer différentes contraintes aux méthodes de compression et évaluent pour chaque ensemble de contraintes (ie. le contexte) leurs performances. L'objectif de ce travail est de déterminer le domaine de prédilection de chacune de ces méthodes afin de pouvoir, par la suite, effectuer dynamiquement la sélection.

Ce document justifie, dans un premier temps, le besoin d'une sélection dynamique. Dans sa seconde partie, il explique les principes de l'algorithme et de la plate-forme de mesures conçue pour l'étalonner. Il donne, enfin, les résultats des mesures de performances qui y ont été menées.

2. Méthodes de compression

Nous avons utilisé pour notre plate-forme trois méthodes de compression choisies parmi les différentes méthodes de compression existantes dans les implémentations actuelles du protocole PPP [SIM 94]. Les méthodes de compression peuvent être classées en deux catégories : les méthodes de compression de contenu qui compressent la totalité du paquet et les méthodes de compression d'en-têtes qui ne compressent que les en-têtes protocolaires.

2.1. Méthodes de Compression de Contenu

2.1.1. Deflate

Deflate est une méthode de compression sans perte qui combine l'algorithme Lempel-Ziv et le codage Huffman. Elle a été définie, à l'origine, par Phil Katz pour la version 2 de son outil de compression PKZip, puis a été standardisée par [DEU 96].

Deflate a été largement utilisée dans les réseaux et la compression de fichiers avant même sa standardisation en 1996. L'algorithme demande un temps de calcul assez important mais donne, en contrepartie, un des meilleurs taux de compression.

L'efficacité de Deflate est comparable aux meilleures méthodes de compression actuelles dans un environnement fiable. Si l'environnement n'est pas fiable, Deflate utilise les mécanismes de recouvrement de CCP [RAN 96a], le protocole de gestion de la compression de PPP, pour se rétablir après une erreur. Ces mécanismes appelés Reset-Request et Reset-Ack induisent des pertes supplémentaires de paquets puisqu'ils réinitialisent complètement la session à chaque fois qu'un paquet est perdu sur le lien. Cela entraîne des délais importants.

2.1.2. Predictor

Predictor, [RAN 96b], est une méthode de compression très rapide quoique moins performante que ses rivales : le taux de compression obtenu en utilisant Predictor n'est pas aussi important que celui de Deflate, par exemple, mais il est plus robuste et plus rapide que ce dernier.

Pour résumer son fonctionnement, le compresseur parcourt la suite d'octets à compresser en essayant de prédire les prochaines séquences, s'il réussit à le faire, il estime que le décompresseur peut le faire aussi. Sinon, il envoie les données brutes.

La spécificité de Predictor réside dans l'absence de gestion d'un historique entre les paquets. Chaque paquet est compressé et décompressé indépendamment des autres. Ceci peut être utile dans les environnements peu fiables puisque la perte d'un paquet n'affecte pas les autres.

2.2. Méthode de Compression d'En-têtes

2.2.1. ROHC

RoHC, RObust Header Compression, a été standardisé par [BOR 01] en 2001 avec un objectif principal : être assez robuste pour être utilisé dans les réseaux cellulaires.

RoHC est modulaire et extensible : il permet l'ajout ultérieur de profils. Un profil étant l'ensemble des mécanismes servant à compresser une suite d'en-têtes particulière (e.g. IP/TCP ou IP/UDP/RTP).

RoHC a aussi une classification très détaillée des champs contenus dans les entêtes en fonction de leur comportement dans le temps. Il classe même certains champs comme CHANGING pour établir leurs comportements de façon dynamique.

Le principal reproche fait à RoHC est sa complexité. La multitude de techniques qui y sont utilisées en fonction de la classification des champs font de lui un outil assez lourd et difficile à implémenter sur les terminaux mobiles ayant des capacités limitées. Par contre, ceci ne devrait pas représenter de problèmes dans le futur étant donné que les capacités de traitement embarquées sont en constante augmentation.

3. Algorithme de sélection dynamique de la méthode de compression

3.1. Idée générale

Chacune des méthodes de compression que nous étudions réagit différemment en terme d'efficacité suivant le contexte dans lequel elle est utilisée. L'idée de l'algorithme de sélection est alors la suivante : à partir d'une évaluation du contexte, la "prédiction" de l'efficacité des différentes méthodes de compression permettra de sélectionner la méthode la plus efficace dans le contexte donné. Nous supposons pour cela que l'efficacité d'une méthode de compression est directement liée au contexte.

$$performances = f(\text{contexte})$$

Bien sûr, la fonction $f()$ est une inconnue, mais nous pouvons, par des moyens expérimentaux, obtenir sa valeur en plusieurs points pour pouvoir, par la suite, les interpoler et obtenir la fonction de prédiction $f_{pred}()$ qui est une approximation de $f()$.

Cette fonction est ensuite fournie à notre algorithme de sélection afin de l'aider à prédire les performances des méthodes de compression et à sélectionner celle qui est la plus adaptée.

Le choix de la méthode doit être dynamique à cause du contexte de mobilité qui ne permettra pas d'avoir des conditions réseau stables. Par exemple, l'algorithme devra choisir de changer de méthode de compression lors d'un handover vertical entraînant des changements de caractéristiques importants.

De plus, étant donnée la polyvalence croissante des terminaux mobiles, il faut s'attendre à ce que les liens sans-fil transportent de plus en plus de flux différents. Tous ces flux ne peuvent pas être compressés de la même manière, d'où la nécessité de différencier les flux.

L'étude est donc organisée en deux parties. Une première partie faisant l'objet de cet article est appelée "phase de mesures". Nous tentons d'y obtenir une fonction $f_{pred}()$ la plus proche possible de $f()$ afin d'avoir une prédiction des performances assez proche de la réalité. Une deuxième partie, s'attachera à la conception de l'algorithme de sélection et proposera une manière d'utiliser la fonction $f_{pred}()$ afin d'élire dynamiquement la méthode de compression la plus adaptée.

3.2. Paramètres de performance considérés

Les critères de performance sont les paramètres qui seront considérés par l'algorithme pour déterminer si une méthode de compression est meilleure qu'une autre. Ainsi, nous avons estimé que les paramètres de performances les plus importants sont :

- τ : Taux de Compression.
- δ : Délai Supplémentaire (temps de traitement).
- π : Taux de Perte Supplémentaire (induit par la méthode de compression).

Le taux de compression, τ , est le paramètre de performance le plus important puisqu'il permet d'évaluer l'efficacité de la compression.

Le délai supplémentaire, δ , est une mesure du temps nécessaire pour l'opération de compression/décompression. Il est important dans la mesure où certaines méthodes, plus rapides que d'autres, sont à privilégier pour les flux ayant des contraintes temps réel.

Le taux de perte supplémentaire, π , permet de privilégier les méthodes de compression dites "robustes". En d'autres termes, les méthodes qui perdent moins de paquets lorsque les conditions réseau sont défavorables.

τ , δ et π ne sont pas les seuls paramètres de performance qui permettent de préférer une méthode à une autre. Nous en avons dénombré d'autres comme la résistance au déséquencement ou encore le délai d'initialisation. Mais, pour le moment, et afin de garder un système simple, nous ne les considérerons pas dans notre étude.

La fonction $f()$, citée précédemment, combine ces trois paramètres de performance pour donner une seule et unique quantification de la performance. Ceci est réalisé grâce à une pondération de chacun de ces paramètres en fonction de l'importance que lui donne l'utilisateur ou l'application qui a généré le flux. Ceci résulte en la formule suivante : $f() = a \times \tau() + b \times \delta() + c \times \pi()$, a,b et c étant fixés par l'utilisateur ou fournis par l'application et peuvent être différents pour chaque type de flux.

3.3. Contexte considéré

En plus des performances, nous devons considérer le contexte dans lequel elles ont été obtenues. Nous avons identifié deux ensembles de caractéristiques : les caractéristiques du réseau et les caractéristiques du flux.

3.3.0.1. Caractéristiques du flux

Les performances de la compression varient significativement en fonction du type de flux à compresser. Trois caractéristiques influencent particulièrement les performances des méthodes de compressions :

- PS : la taille des paquets,

- IPD : le délai inter-Paquets,
- DR : le taux de redondance des données.

La taille des paquets, PS, permettra de départager, les méthodes de compression de contenu et celles qui ne compressent que les entêtes, étant donné que ces dernières sont pénalisées par les paquets volumineux.

Le délai inter-paquets, IPD, est une mesure destinée à pénaliser les méthodes avec un rafraîchissement de contexte périodique quand le délai entre les paquets est très espacé. Ceci est le cas notamment de RoHC en mode unidirectionnel. En effet, celui-ci envoie, à intervalles réguliers, des paquets non compressés afin de régénérer son contexte.

Le taux de redondance, DR, différencie les flux dont le contenu est d'ores et déjà compressé et ceux qui contiennent des informations redondantes pouvant être aisément compressées. Ce paramètre permettra d'avantager les méthodes de compression d'en-têtes dans le cas d'un flux déjà compressé dans la mesure où celles-ci ne tentent pas de recompresser tout le paquet mais seulement les en-têtes et obtiennent ainsi de meilleurs résultats. En effet, lorsque les données transmises sont déjà compressées (image jpeg, fichier ZIP, ...), tenter de les recompresser entraîne souvent une augmentation de la taille des paquets au lieu de la diminuer.

3.3.0.2. Caractéristiques du réseau

Les caractéristiques du lien traversé influent aussi sur l'efficacité des méthodes de compression. Cette influence se ressent surtout au niveau du paramètre de performance relatif à la robustesse de la méthode : π . Mais cela se répercute aussi, quoique de façon moins directe, sur le taux de compression. Nous avons choisi de limiter notre étude aux deux contraintes réseau suivantes :

- PL : le taux de perte en paquets du lien.
- OWD : le délai unidirectionnel.

Quand le taux de perte sur le lien, PL, est élevé, certaines méthodes de compression sont complètement désynchronisées et passent beaucoup de temps à essayer de se rétablir. Considérer cette contrainte permettra d'avantager les méthodes dites robustes lorsque le taux de perte vient à devenir important.

Enfin, Le délai unidirectionnel, OWD, permettra de considérer le temps nécessaire à une opération de réparation du contexte mis en place par la méthode de compression. Ceci donnera l'avantage aux méthodes utilisant des techniques de prévention de désynchronisation lorsque le délai aller-retour est important.

3.4. Campagne de mesures

L'objectif de la campagne de mesures est d'obtenir les fonctions de prédiction $\tau_{pred}()$, $\delta_{pred}()$ et $\pi_{pred}()$ les plus proches possible de la réalité. Puisque ces fonctions

sont obtenues par interpolation, une condition de leur exactitude réside dans le nombre de points qui interviennent et dans l'uniformité de leur distribution. En pratique, ceci se traduit par un nombre important d'expériences menées, à chaque fois, sous des contextes différents. Nous avons développé une plate-forme de mesures permettant d'assurer, à la fois, la diversification des flux et des caractéristiques du lien.

4. Plate-forme de mesures

Fig.1 illustre la plateforme actuellement installée dans notre laboratoire. On y distingue une partie mobile schématisée par un véhicule et une partie fixe incluant les serveurs. Ces deux réseaux IPv6 sont reliés à travers la connectivité IPv4 fournie par le réseau de transit intermédiaire. Ceci est réalisé à l'aide d'un mécanisme de tunneling (L2TP dans notre cas) qui encapsule la totalité d'un paquet IPv6 dans un paquet IPv4.

Lors d'une expérience, les deux extrémités du tunnel effectuent l'opération de compression/décompression en générant des traces des paquets empruntant le tunnel. Par la suite, ces traces sont analysées pour dégager, d'une part, les valeurs des contraintes sous lesquelles l'expérience a été menée et, d'autre part, une évaluation des performances produites par la méthode de compression.

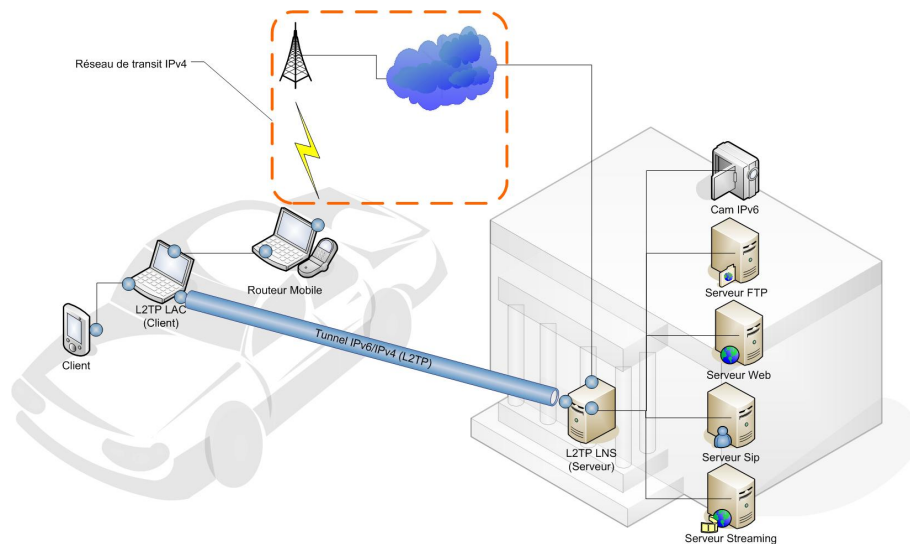


Figure 1. Architecture de la plateforme

4.1. Piles protocolaires

Les piles protocolaires relatives à chacune des machines de la plate-forme sont présentées dans fig.2. Les deux extrémités (le serveur et le client) ont une pile IPv6 classique. Les extrémités du tunnel sont des routeurs dits à double pile IPv4/IPv6. Ils encapsulent chaque paquet IPv6 reçu dans un paquet PPP/L2TP/UDP/IPV4 après l'avoir compressé. Le paquet peut, par la suite, être routé en IPv4 dans le réseau de transit intermédiaire. Les paquets peuvent éventuellement passer à travers des passerelles et des firewalls puisque L2TP fonctionne au dessus de UDP. Cela nous permettra, par exemple, de conduire des expérimentations sur des réseaux GPRS ou UMTS.

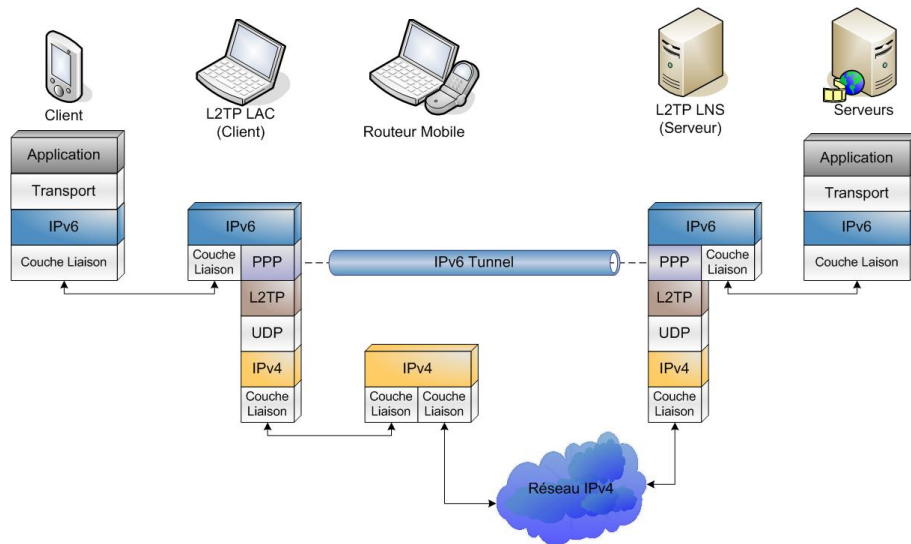


Figure 2. Piles protocolaires

Grâce à cette architecture, nous pouvons connecter virtuellement tout client et tout serveur sans aucune contrainte autre que l'usage d'IPv6. Durant une expérience, des traces du passage des paquets sont collectées et analysées afin d'extraire les différentes mesures nécessaires à l'étalonnage de l'algorithme de sélection.

5. Résultats obtenus

Les résultats obtenus tentent d'établir les relations qui peuvent exister entre le contexte et les paramètres de performance étudiés.

5.1. Variation du taux de perte de la compression (π)

Le nuage de points représenté dans fig.3 décrit comment les performances des protocoles de compression étudiés varient en fonction du taux de perte du lien. Nous remarquons que RoHC ne réagit pas au taux d'erreurs de cet ordre puisque celui-ci a été prévu pour fonctionner sur des liens peu fiables. Par contre, les taux de perte de Deflate et Predictor varient de façon sensible avec le taux de perte du lien.

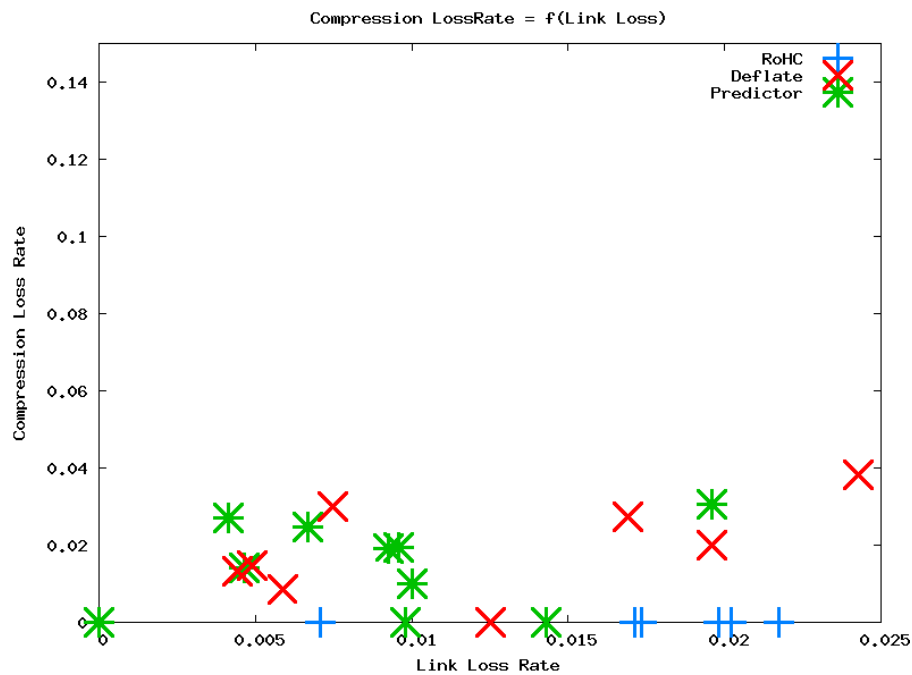


Figure 3. Variation du taux de perte dû à la compression

5.2. Variation du temps de traitement (δ)

Le temps de traitement est un paramètre important dans le cas de flux temps-réel comme la voix par exemple. Fig.4 représente la variation de δ en fonction de la taille des paquets. Sans surprise, Deflate prend beaucoup plus de temps à traiter les données que Predictor qui ne semble pas subir l'influence de la taille des paquets. RoHC prend encore plus de temps que Deflate puisqu'il a atteint des pics à 120 ms. Ceci peut s'expliquer par sa complexité relative et par la multitude de techniques qu'il intègre.

Il faut cependant noter que ces mesures ont été effectuées sur des machines ayant des capacités de traitement relativement faibles et que les différences n'auraient pas été aussi importantes sur des machines plus puissantes. De plus, l'implémentation de

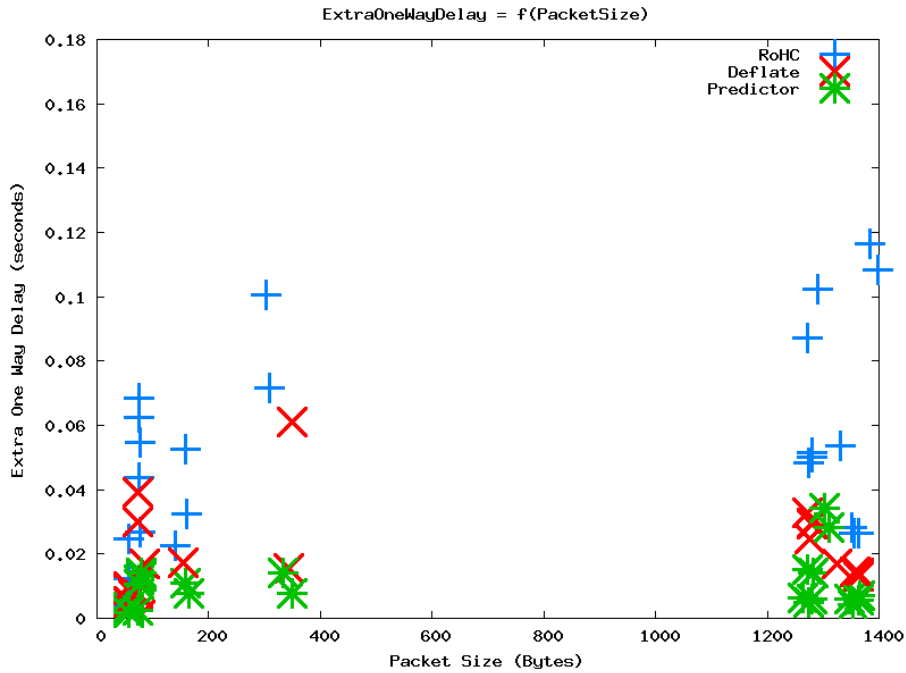


Figure 4. Variation du temps de traitement

RoHC utilisée, [MIN 03], est une implémentation expérimentale développée au sein de notre laboratoire.

5.3. Variation du Taux de Compression (τ)

Fig.5 représente la variation du τ de chacune des trois méthodes des compression étudiées en fonction des contraintes “taille du paquet” et “redondance des données”.

Nous pouvons en conclure RoHC se comporte de façon prévisible. Il compresse les paquets peu volumineux ayant un contenu déjà compressé beaucoup mieux que les méthodes de compression de contenu. Quand le taux de redondance des paquets vient à s'élever, c'est Deflate qui prend le dessus. Predictor, quant-à lui, suit Deflate de très près et ne le dépasse que rarement et de très peu notamment dans la zone où il y a des paquets volumineux et pré-compressés. Dans cette zone, nous remarquons aussi la faible valeur ajoutée de la compression puisque les trois méthodes donnent un taux de compression inférieur à 10%. Ceci pourrait justifier la décision de désactiver complètement la compression pour ce type de flux.

Cette courbe donne une idée générale du fonctionnement de l'algorithme de sélection. Pour être plus précis, il faut imaginer la représentation de la fonction de pré-

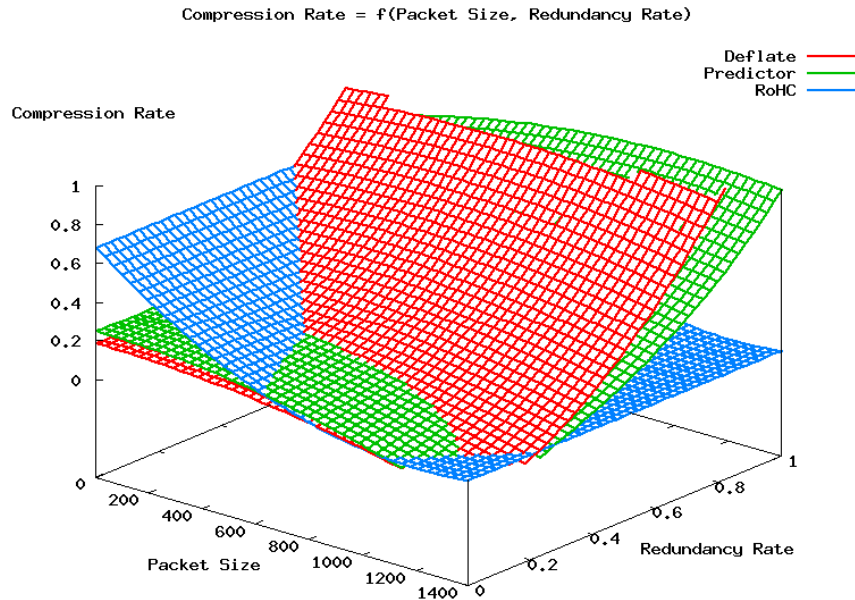


Figure 5. Variation du Taux de Compression

diction $f_{pred}()$ en fonction des cinq contraintes de notre système. L'ajout des autres paramètres pourrait augmenter le contraste entre les courbes et avantager Notamment RoHC étant donné qu'il est le plus robuste.

L'algorithme de sélection disposant d'une telle fonction, peut parcourir l'espace des contraintes et choisir à chaque instant la méthode correspondant à la courbe la plus élevée.

6. Conclusion

L'objectif ultime de notre travail est de concevoir un algorithme de sélection dynamique de la méthode de compression. Une étape clé de la conception consiste en la collecte des mesures servant à fournir des indicateurs pour permettre la prédiction des performances et la prise de décision.

La campagne de mesures menée a aussi servi à montrer les différences qui existent entre les méthodes de compression et leur inefficacité dans certains contextes justifiant ainsi le besoin d'un mécanisme de sélection dynamique de la méthode de compression.

Nous avons établi la supériorité des méthodes de compression de contenu (Deflate en particulier) lorsque les flux ne sont pas compressés et que les paquets sont assez volumineux. De même, nous avons montré l'avantage de la compression d'en-têtes lorsque les flux sont pré-compressés et que les paquets sont de taille réduite. L'étude a également mis en lumière l'inégalité des méthodes de compression en termes de temps de traitement et de résistance aux erreurs.

Notre plate-forme pourra servir à situer les performances des nouvelles méthodes de compression par rapport aux méthodes préexistantes. Elle pourra aussi servir à aider dans le choix statique d'une méthode de compression dans un contexte stable.

Dans l'avenir d'autres méthodes de compressions vont être intégrées dans la plate-forme à des fins d'évaluation. Ensuite, nous envisageons la considération d'autres contraintes et d'autres paramètres de performance. Enfin, nous projetons d'effectuer une étude statistique des résultats pour établir des corrélations entre les paramètres de performance et les contraintes étudiées.

7. Bibliographie

- [BOR 01] BORMANN, BURMEISTER C., DEGERMARK M., FUKUSHIMA H., HANNU H., JONSSON L.-E., HAKENBERG R., KOREN T., LE K., LIU Z., MARTENSSON A., MIYAZAKI A., SVANBRO K., WIEBKE T., YOSHIMURA T., ZHENG H., « Robust Header Compression (ROHC) : Framework and four profiles : RTP, UDP, ESP, and uncompressed », *Internet Request for Comments*, , n° 3095, 2001.
- [DEU 96] DEUTSH P., « PPP Deflate Protocol », *Internet Request for Comments*, , n° 1979, 1996.
- [MIN 03] MINABURO A., « Compression des en-tetes sur les réseaux bas-débit », , 2003.
- [RAN 96a] RAND D., « The PPP Compression Control Protocol (CCP), STD 51, RFC 1962 », 1996.
- [RAN 96b] RAND D., « PPP Predictor Compression Protocol », RFC n° 1978, août 1996.
- [SIM 94] SIMPSON W., « The Point-to-Point Protocol (PPP), STD 51, RFC 1661 », 1994.