



HAL
open science

Etudes de paramètres de techniques de compression pour la conception d'un algorithme de sélection

Souha Bannour, Abdelfettah Belghith, Jean-Marie Bonnin

► **To cite this version:**

Souha Bannour, Abdelfettah Belghith, Jean-Marie Bonnin. Etudes de paramètres de techniques de compression pour la conception d'un algorithme de sélection. SETIT 2007 : 4th international conference: Sciences of electronic, technologies of information and telecommunications, Mar 2007, Tunis, Tunisie. hal-02900803

HAL Id: hal-02900803

<https://hal.science/hal-02900803>

Submitted on 16 Jul 2020

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.

Etudes de paramètres de techniques de compression pour la conception d'un algorithme de sélection

Souha BANNOUR^{*}, Abdelfattah BELGHITH^{*} et Jean-Marie BONNIN^{**}

^{}Laboratoire KRISTAL, Ecole Nationale des Sciences de l'Informatique (Tunisie)*

bannour_souha@yahoo.fr
abdelfattah.belghith@ensi.rnu.tn

*^{**}Laboratoire RSM, Ecole Nationale Supérieure des Télécommunications Bretagne (France)*

jean-marie.bonnin@irisa.fr

Résumé: Le passage du protocole IPv4 au protocole IPv6 n'est pas encore totalement réalisé. Dans le but d'envoyer des paquets IPv6 indépendamment du type du réseau, nous avons eu recours au mécanisme de tunneling L2TP (*Layer 2 Tunneling Protocol* : ensemble des processus d'encapsulation, de transmission et de désencapsulation des données)..

Or, le succès de déploiement du protocole IPv6 dépend étroitement du coût imposé par l'opérateur sur l'utilisateur. Pour minimiser ce coût et améliorer les mécanismes de tunneling développés actuellement, nous devons essayer de limiter l'utilisation de la bande passante. Pour cela, nous avons utilisé la compression.

Cet article présente une étude des différents paramètres de certains algorithmes de compression ainsi que les caractéristiques du réseau pour divers types de flux en vue de concevoir un algorithme intelligent de sélection de technique de compression.

Nous avons montré que l'efficacité d'une technique de compression varie suivant plusieurs paramètres.

Mots clés: IPv6, mécanisme de tunneling, coût, bande passante, algorithme de compression, sélection.

INTRODUCTION

L'évolution actuelle des services multimédias tend à répondre au besoin de la mobilité chez les utilisateurs. Cette tendance a abouti à l'échange d'informations entre différents types de réseaux de communication (Internet et Réseaux téléphoniques). Les informations à échanger peuvent avoir différentes caractéristiques et chaque réseau les traite différemment. Le réseau Internet et les réseaux téléphoniques mobiles se confondent de plus en plus pour donner un service homogène de bout en bout.

Dans le but d'assurer de meilleures performances dans les réseaux bas débit, nous avons pensé à optimiser l'utilisation de la bande passante tout en garantissant la qualité souhaitée par l'utilisateur. D'où l'idée de recourir aux techniques de compression. Cependant, les questions qui se posent, dans ce cas, sont essentiellement : quelles techniques de compression? et pourquoi?

Les diverses techniques de compression offrent différentes performances en terme d'espace, temps et taux de compression selon les caractéristiques du réseau et celles du trafic à transmettre.

En général, le choix de la technique de compression, dans le cas de l'existence de plus d'un algorithme, se négocie au cours de l'établissement de la connexion.

Plusieurs équipes de recherche ont travaillé sur l'adaptation des paramètres d'un algorithme de compression selon des critères particuliers et ceci surtout pour la compression des images et de la voix sur IP [Smi 95] [Tay 01].

Cependant, notre travail ne se limite pas à un seul algorithme de compression. Tel est le cas d'une équipe de recherche à l'école des Mines Pont de Paris dont l'étude s'est basée sur un ensemble d'algorithmes de compression d'images [Gil 98].

Tout au long de cet article, nous allons présenter la compression dans les réseaux ainsi que son intérêt. Nous enchaînons par la description de notre solution. Nous abordons, par la suite, l'aspect technique de notre travail. Enfin, nous analysons et commentons quelques résultats obtenus.

1. La compression dans les réseaux

Etant donné que notre but consiste essentiellement à optimiser l'utilisation de la bande passante tout en

garantissant la qualité souhaitée par l'utilisateur, nous avons eu recours à la compression avec l'idée de concevoir un algorithme de sélection de techniques de compression.

La compression a vu le jour depuis longtemps pour remédier au problème de débit offert par les modems. Certes, ce besoin a diminué avec l'évolution des réseaux filaires et la naissance de nouvelles technologies telle que la fibre optique, mais il a surgit de nouveau avec les réseaux sans fil, étant donné que les ressources radio sont rares et partagées.

La compression est généralement effectuée dans la couche liaison. Elle pourrait être implémentée ailleurs, mais c'est au niveau 2 qu'elle est la plus efficace puisqu'elle peut aussi compresser les en-têtes des couches supérieures.

Pour pouvoir compresser, il faut une connexion point à point ou une émulation de liaison point à point au-dessus de l'internet. Donc, soit on compresse sur un lien simple soit on compresse au-dessus d'un tunnel comme le cas de L2TP.

Nous pouvons diviser les méthodes de compression en deux catégories, à savoir la compression de données et la compression d'en-tête.

La compression de données considère le paquet comme un tout (en-têtes + contenu) et le compresse de manière globale. Ce type de compression convient parfaitement lorsque l'en-tête du paquet est de taille négligeable par rapport à sa charge. Cependant, dans le cas d'un paquet non volumineux, cette technique ne compresse quasiment pas ou étend même le paquet dans certains cas.

Par contre, la compression d'en-têtes dissocie le paquet en en-têtes et contenu et compresse uniquement les en-têtes. Elle cible les flux à caractère interactif et temps réel telle que la voix ainsi que les flux vidéo. Cette technique obtient de bons résultats sur de tels flux car la charge du paquet est minime en comparaison avec l'en-tête qu'il génère. On peut donc réduire l'overhead de façon spectaculaire en compressant les en-têtes. Tel est le cas des mécanismes de tunneling qui doivent disposer d'une compression d'en-têtes afin d'atténuer la sur-charge due à l'encapsulation des paquets IPv6 dans des paquets IPv4.

2. Solution proposée

Le but de notre travail est d'assurer de meilleures performances et d'améliorer la qualité de transmission des données sur les réseaux bas débits. Les diverses techniques de compression offrent différentes performances en terme d'espace, temps et taux de compression, selon les caractéristiques du réseau et celles du trafic à transmettre.

Quand on a plusieurs types de données à véhiculer, le choix d'une technique de compression au lieu d'une autre peut ne pas être bénéfique. D'où l'idée de concevoir un algorithme de sélection de techniques de

compression qui permet de choisir dynamiquement la technique de compression la plus adéquate au flux à transmettre.

Donc, étant donné un ensemble de méthodes de compression et un contexte d'exécution, quelle est la méthode de compression la plus adaptée au contexte ?

2.1. Principe de l'algorithme de sélection

Notre travail consiste à étudier certains paramètres d'algorithmes de compression ainsi que les caractéristiques du réseau pour divers types de flux en vue de concevoir un algorithme de compression intelligent qui consiste à choisir la technique de compression adéquate pour chaque flux selon le type de données et l'objectif de l'utilisateur ou de l'application. Cette sélection doit se baser sur un ensemble d'algorithmes de compression stockés dans une base de données, les caractéristiques de ces algorithmes (type, complexité. . .), les caractéristiques des divers types de flux pouvant être transmis. . .

En effet, étant donné un ensemble de spécifications pour l'application cible, l'algorithme de sélection a pour rôle de choisir, parmi les techniques de compression disponibles, la technique la plus convenable en se référant aux buts souhaités et aux contraintes à respecter à partir d'un ensemble de techniques offertes comme le montre la figure1 ci-dessous.

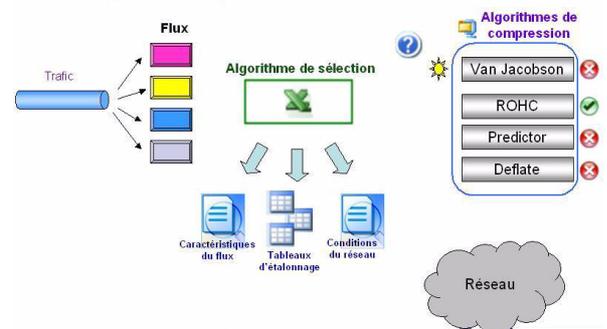


Figure 1.Principe de l'algorithme de sélection

Ceci permet d'assurer de meilleures performances, à savoir, une utilisation plus optimisée de la bande passante et un gain en temps de transmission.

En général, avoir plusieurs techniques qui offrent différents degrés de performances est plus efficace comme solution.

Afin que l'algorithme de sélection soit bien conçu, nous nous sommes basés sur des mesures réelles et de vrais exemples.

2.2. Quelles techniques de compression ?

Une question importante se pose, dans le cas d'algorithme de sélection, concernant le nombre de techniques de compression qui peuvent être retenues comme candidates pour réaliser les fonctionnalités

souhaitées.

Pour cela, nous allons nous baser sur 2 types de techniques :

- Les techniques qui compriment le paquet IP et ne maintiennent pas d'état.
- Les techniques qui compriment seulement les en-têtes mais doivent maintenir un contexte.

Etant donné que nous travaillons sous le système d'exploitation Unix, plus précisément la distribution FreeBSD 5.3, nous avons utilisé les techniques de compression qui y sont implémentées, à savoir Van Jacobson (la première technique de compression d'en-tête) comme technique de compression d'en-tête, deflate (une des meilleures méthodes en terme de taux de compression mais très sensible aux erreurs) et predictor (compresse moins, mais résiste plus aux erreurs) comme techniques de compression de données. En plus de ces techniques, nous avons eu recours à la technique de compression d'en-tête ROHC qui a été implémentée par une équipe de recherche à l'ENST Bretagne [Car 03].

Sachant que chaque technique de compression a été conçue dans le but de répondre à un besoin bien déterminé, ces algorithmes sont caractérisés par des options différentes.

2.3. Quels types de flux ?

La deuxième question à laquelle il faut répondre est : Quels types de flux utiliser ?

Nous avons eu recours au flux vidéo (Video Streaming), la voix sur IP (VoIP : Voice over IP) et le transfert de fichiers (ftp : file transfert protocol).

2.4. Critères de sélection

Une troisième question s'impose : Quels sont les critères de sélection sur lesquels on peut se baser ?

Les critères de sélection dépendent essentiellement du type de flux. Parmi ces critères, on distingue :

- Les caractéristiques du flux : c'est à dire RTP ou TCP
- Les caractéristiques du réseau : tel que le taux d'erreur ou le type de la liaison
- Le taux de compression : le rapport entre la taille des informations non comprimées et leur taille une fois qu'elles ont été compressées. Ce paramètre permet de mesurer la performance de l'algorithme de compression utilisé.
- Le délai : c'est le temps de transmission de bout en bout (OWD : One Way Delay)
- Le temps d'interarrivée ou gigue : les applications telles que le son et la vidéo sont très sensibles au temps écoulé entre arrivées successives des données. Sans un débit de données régulier, le son risque d'être haché.
- La bande passante : C'est la capacité d'une liaison de données, déterminée par les technologies de transmission mises en oeuvre à l'aide des équipements situés à chaque extrémité de cette liaison.
- Le débit : représente la quantité d'informations

empruntant un canal de communication selon un intervalle de temps donné. Le débit d'une connexion internet s'exprime généralement en kbps (kilobit par seconde).

Le choix entre techniques de compression de données et celles d'en-tête est aussi important étant donné que dans ce dernier cas, la sensibilité aux pertes et aux délais trop importants est probablement beaucoup plus forte.

3. Plate-forme de mesures

Nous avons travaillé sur le système d'exploitation FreeBSD parce qu'en plus des techniques de compression implémentées au sein du protocole PPP de chaque système, une implémentation de ROHC sur FreeBSD a été mise en place par une équipe de recherche à l'ENST Bretagne [Car 03]. En outre, FreeBSD possède une application PPP utilisateur ce qui n'est pas le cas des systèmes Linux.

Etant donné que notre travail se situe dans le contexte d'amélioration des performances du mécanisme de tunneling L2TP, nous avons commencé par l'installation de ce protocole. Or, toutes les versions de L2TP qui existaient quand on avait débuté notre travail, sont incomplètes et n'assurent pas le fonctionnement complet d'un tunnel.

Nous avons alors été ramenés à chercher un protocole dont le mécanisme est semblable à celui de L2TP et ayant le même principe de fonctionnement. PPPoE était la solution. Nous avons alors mis en place

- Un serveur PPPoE pour permettre l'établissement d'une connexion PPP.
- Un client PPP qui sollicite l'établissement d'une liaison point à point.

Après avoir choisi les types de flux sur lesquels nous allons effectuer nos mesures, nous avons installé :

- Un serveur de flux : Le serveur permet de transmettre des flux.
- Un processus client: Le client se connecte au serveur et sollicite l'émission d'un flux donné.

Ceci pour le protocole UDP.

Nous devrions aussi effectuer des mesures pour le protocole de transport TCP et nous avons choisi de reproduire, comme usage de ce protocole, le transfert de fichiers, nous avons alors installé :

- Un serveur ftp.

4. Choix des paramètres

La première chose à fixer est : Qu'est ce que nous allons mesurer ? Afin de répondre à cette question, il faut préciser le but du travail à réaliser. En effet, notre but est de minimiser le temps de transmission et l'utilisation de la bande passante tout en garantissant une bonne qualité de réception des données, et ce en

utilisant la technique de compression convenable à chaque type de flux à transmettre. Pour ceci, nous avons eu l'idée de tester les techniques de compression implémentées au sein de PPP.

Afin d'observer les performances de ces techniques, nous allons mesurer le taux de compression obtenu par chacune d'elle et par les divers combinaisons existantes entre elles en les appliquant sur différents types de flux. Le taux de compression c'est le rapport entre la taille des informations non comprimées et leur taille une fois qu'elles ont été compressées. Ce paramètre permet de mesurer la performance de l'algorithme de compression utilisé

Afin d'évaluer les performances d'un système, le temps est l'un des facteurs essentiels à tenir en compte surtout quand la facture est fonction du temps. Le délai de transmission est l'un des paramètres de performance les plus significatifs, c'est la durée de temps pendant laquelle un paquet parvient à destination (temps qui sépare l'envoi du premier bit et la réception du dernier bit), c'est ce qu'on appelle le OWD (One Way Delay).

En outre, nous avons mesuré le temps d'interarrivée entre les paquets, le temps de traitement ainsi que le taux de perte des paquets.

5. Architecture du réseau de mesures

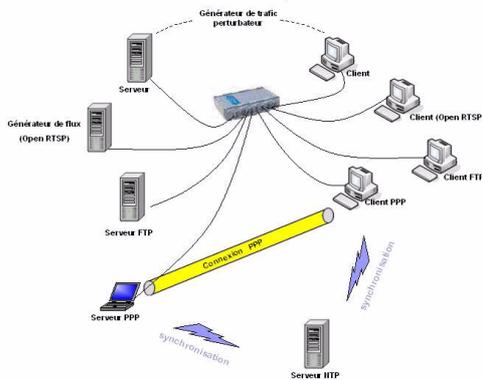


Figure 2. Plate-forme de mesures

La figure2 ci-dessus illustre l'architecture du réseau de mesures que nous avons déployée. Cette architecture est constituée d'un mini-hub (de bande passante égale à 10 Mb) lié au réseau de l'ENSi et 3 PC (les 2 extrémités de la liaison PPP et le serveur de temps pour la synchronisation).

Deux autres machines nous ont servi pour générer un trafic perturbateur sur le réseau dans le but de minimiser la bande passante disponible et voir l'effet de cette variation sur les diverses techniques de compression et le taux de perte de paquets dans le cas où cette variation cause des pertes.

6. Description de l'expérience

Avant de commencer les mesures, il faut vérifier

que les deux extrémités sont bien synchronisées. Puis, nous procédons comme suit :

Au départ, nous activons les techniques de compression souhaitées qui doivent être supportées par le système. Puis, nous établissons une liaison PPPoE entre les deux extrémités. Ensuite, nous lançons l'openRTSP chez le serveur de flux afin qu'il soit prêt à envoyer les paquets (mode serveur RTSP) et nous faisons appel aux données souhaitées qui correspondent au type activé chez le serveur, depuis le client. En même temps, nous activons le topdump sur les deux machines afin de capturer les paquets échangés tout en mentionnant les critères désirés tels que l'interface de capture, le type de paquets à sniffer et le nombre d'octets à capturer.

Après ces prises de mesures, nous observons les résultats grâce à ethereal [Eth 04] qui permet de visualiser les valeurs numériques de façon plus lisible et de détailler les différents champs des paquets capturés, nous traitons les fichiers de statistiques obtenus et nous dégageons les résultats dont on a besoin pour tracer, à la suite, les courbes désirées avec gnuplot [Gnu 04] et Microsoft Excel.

Enfin, nous analysons et interprétons les courbes obtenues. Ces différentes étapes sont illustrées dans la figure 3.

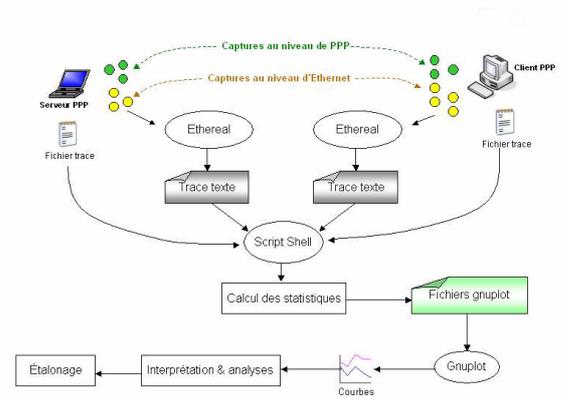


Figure 3. Procédure de prise et de traitement des mesures

7. Résultats obtenus

Nous avons fait des mesures pour les deux types de protocoles de transport les plus utilisés dans l'Internet, à savoir UDP et TCP. Nous avons ainsi reproduit essentiellement deux usages courants à savoir : la voix sur IP (UDP) et les flux vidéos (UDP) et nous avons abordé de façon superficielle le transfert de fichiers (FTP sur TCP).

Pour les flux vidéo, nous avons eu recours à des séquences de divers type, à savoir : MP3, Transport Streaming, MPEG1_Vidéo, MPEG1_Audio-Vidéo, MPEG2_Vidéo et MPEG2_Audio-Vidéo.

Les figures 4 et 5 ci-dessous permettent de mieux comprendre le taux de compression calculé. En effet, elles illustrent les parties compressées par les

techniques de compression implémentées dans PPP dans le cas de la compression de données et celle d'en-tête.

- Compression de données

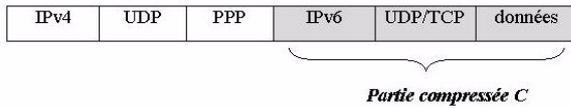


Figure 4. La partie compressée, du paquet véhiculé dans le cas de la compression de données

C représente la partie qui sera compressée dans ce cas.

Soit 'A' la taille de la partie C avant compression et 'a' la taille de cette partie après la compression => le taux de compression T est égal au quotient de A par a ($T = A/a$).

- Compression d'en-têtes

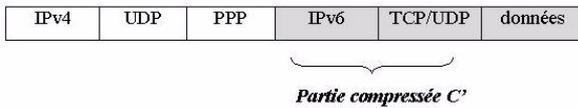


Figure 5. La partie compressée, du paquet véhiculé dans le cas de la compression d'en-tête

C' représente la partie qui sera compressée dans ce cas.

Soit 'A' la taille de la partie C' avant compression et 'a' la taille de cette partie après la compression => le taux de compression T est égal au quotient de (A + données) par (a + données) ($T = (A + données) / (a + données)$).

Nous avons réalisé les mesures sur trois étapes :

7.1. PPP sans erreurs

Au départ, nous avons effectué nos mesures en injectant le trafic souhaité sous les conditions normales du réseau (sans génération d'erreurs ni de flux perturbateur). A chaque fois, nous changeons de technique de compression et nous prenons les mesures nécessaires pour tracer les courbes.

Prenons l'exemple du flux de type MPEG1-Audio Vidéo (1AV).

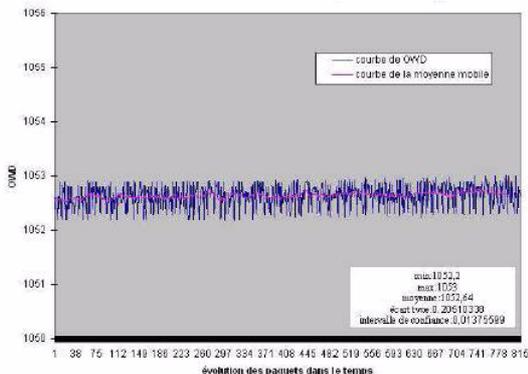


Figure 6. Variation du OWD dans le cas de deflate pour un flux de type 1AV

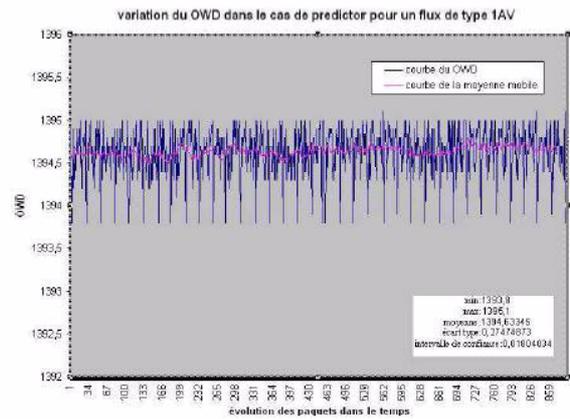


Figure 7. Variation du OWD dans le cas de predictor pour un flux de type 1AV

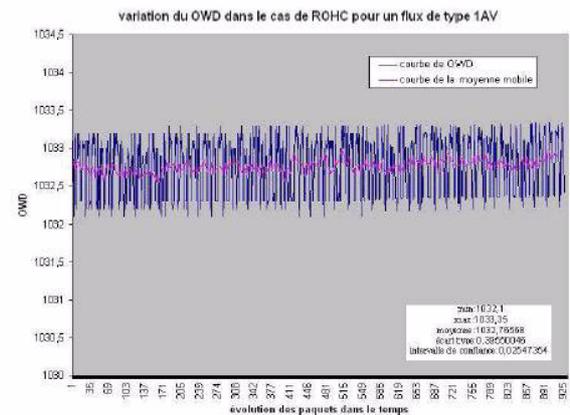


Figure 8. Variation du OWD dans le cas de ROHC pour un flux de type 1AV

Nous remarquons que, pour les trois techniques, les courbes de OWD ne varient pas beaucoup par rapport à la courbe de la moyenne mobile et par valeurs des écarts types sont faibles.

A partir des résultats affichés dans les courbes précédentes, nous constatons que la technique de compression ROHC offre le meilleur temps de transmission pour un flux de type 1AV (en moyenne égal à 1032,766 ms par rapport à 1052,2 pour deflate et 1394,633 dans le cas de predictor).

Comparons les taux de compression ainsi que le temps de compression (en ms) de ces 3 techniques.

	deflate	predictor	ROHC
min	0,0347	0,1779	0,5472
max	0,9985	1	1,0556
moyenne	0,708	0,806	0,930
écart type	0,2986	0,2455	0,0720
intervalle de confiance	0,0577	0,0043	0,0018

Tableau 1. Tableau comparatif des taux de compression pour un flux de type 1AV

	deflate	predictor	ROHC
min	2,7	0,3	2
max	16,9	22,3	15,7
moyenne	7,4757	5,311	6,8153
écart type	3,3789	4,3133	3,0983
intervalle de confiance	0,7699	0,3131	0,559

Tableau 2. Tableau comparatif des temps de compression pour un flux de type 1AV

Notons que le temps de décompression est 2 fois plus grand que le temps de compression dans le cas de deflate et predictor. Il est légèrement supérieur au temps de compression dans le cas de ROHC.

C'est vrai que ROHC n'offre pas les meilleurs taux de compression, mais il est le plus stable et ne varie pas beaucoup autour de sa moyenne (d'après les valeurs de l'écart type ainsi que l'intervalle de confiance du tableau). Cette technique de compression est aussi la plus stable de point de vue temps de traitement. En outre, elle offre les meilleurs temps d'interarrivée avec une moyenne égal à 25,22 ms.

Les calculs ont montré que le temps de décompression pour ROHC ne diffère pas beaucoup de son temps de compression (dans ce cas, il est égal en moyenne à 7,02 ms);

Par contre, les techniques de compression de données deflate et predictor mettent plus de temps à décompresser les paquets qu'à les compresser (par exemple, la moyenne de temps de décompression de deflate dans ce cas est égale à 11,98 ms).

On a déjà vu que ROHC présente un temps de transmission inférieur aux autres. D'où, nous pouvons conclure que pour un flux de type MPEG1-Audio Vidéo, la technique de compression d'en-tête ROHC est la plus performante.

Or, la technique de compression ROHC est une solution payante qui n'est pas encore disponible avec les distributions d'UNIX. Ainsi, si les deux bouts disposent de cette technique, ils ont intérêt à l'utiliser puisqu'elle est la plus performante, d'après les résultats que nous avons observés; Sinon, il faudrait choisir entre deflate et predictor.

D'après les figures 6 et 7, nous remarquons que la technique de compression deflate offre de meilleurs temps de transmission. Par contre, cette technique présente une moyenne de temps de traitement supérieure à celle de predictor. Mais si nous considérons le temps total (traitement+transmission), deflate s'avère le plus performant. En outre, il n'y a pas une grande différence entre ces deux techniques de compression de point de vue taux de compression et de point de vue temps d'interarrivée dont la moyenne est égale à 25,88 ms pour deflate et 24,69 ms pour predictor. D'où, nous pouvons conclure que deflate est meilleur que predictor.

Nous avons procédé de la même manière avec les autres types de données et nous avons obtenu le

tableau récapitulatif suivant :

	technique de compression	en cas de non disponibilité de ROHC
1AV	ROHC	deflate
1V	ROHC	predictor
2AV	ROHC	deflate
2V	deflate	deflate
TS	predictor	predictor
MP3	predictor	predictor
VoIP	predictor	predictor
TCP (fichier word)	deflate	deflate
TCP (fichier MP3)	ROHC	predictor

Tableau 3. Tableau récapitulatif des techniques de compression les plus convenables

7.2. PPP avec erreurs

Dans son travail de thèse, Ana Carolina a montré que ROHC est assez robuste et n'est pas sensible aux erreurs. Cette technique a la capacité de supporter sans perdre le contexte: la propagation d'erreurs, le taux des erreurs bits (BER : Bit Error Rate) et des délais aller-retour variables (RTT : Round Trip Time).

Nous avons conçu un générateur d'erreurs permettant d'avoir les taux d'erreurs par bits suivants : 10^{-5} , 10^{-4} , 20^{-5} et 50^{-4} afin d'observer le pourcentage de paquets perdus qu'on peut obtenir dans ces divers cas en appliquant les techniques de compression de données deflate et predictor.

Nous avons noté que pour un BER égal à 10^{-4} , la connexion PPP est interrompue car la qualité de transfert est très mauvaise.

Nous avons alors dressé un tableau récapitulatif (tableau 4) des pourcentages de paquets perdus dans le cas des techniques de compression de données deflate et predictor pour les divers types de flux et pour les BER suivants : 10^{-5} , 20^{-5} et 50^{-4} ,

		10^{-5}	20^{-5}	50^{-4}
1AV	deflate	69,35	68,83	88,57
	predictor	67,21	68,42	83,65
1V	deflate	37,81	65,96	84,07
	predictor	42,75	65,54	87,8
2AV	deflate	68,12	88,51	94,51
	predictor	71,34	87,22	92,31
2V	deflate	44,1	50,99	82,82
	predictor	45,47	57	80,71
TS	deflate	63,2	67,09	72,38
	predictor	64,01	73,22	79,08
MP3	deflate	9,41	18,29	32,74
	predictor	15,24	21,12	36,03
VoIP	deflate	10,26	30,24	56,78
	predictor	16,09	40,05	60,39

Tableau 4. Pourcentage de perte des paquets pour les divers flux

Sachant que, sans compression, les moyennes des taux de pertes des paquets observées sont notées dans le tableau 5 ci-dessous.

	10^{-5}	20^{-5}	50^{-4}	10^{-4}
1AV	3,47	6,49	14,63	26,89
1V	3,4	6,29	14,27	26,96
2AV	1,81	3,75	8,15	16,67
2V	1,82	3,68	8,39	16,93
TS	10,38	16,47	38,22	63,06
MP3	8,46	15,03	33,5	54,96
VoIP	8,72	16,48	37,33	61,86

Tableau 5. Taux de perte des paquets pour un flux non compressé

Nous remarquons que la perte de paquets engendrée par un taux d'erreur bits est très élevée. Donc, on n'a pas intérêt utiliser une technique de compression de données surtout que, dans le cas où on ne compresses pas les données, le pourcentage de paquets perdus est beaucoup inférieur (sauf dans le cas d'un flux de type MP3 pour un BER égal à 50^{-4} où la différence de pourcentage de pertes n'est pas grande : 33,5% sans compression et 32,724 en utilisant la compression deflate). Les techniques de compression deflate et predictor sont très sensibles aux erreurs.

7.3. Variation de la Bande Passante (BP)

Afin de voir l'impact de la taille de la bande passante, nous avons varié cette taille grâce à la génération d'un flux perturbateur sur le réseau.

La bande passante est initialement égale à 10Mb. Nous l'avons limité en générant des flux CBR (Constant Bit Rate) et nous avons observé le taux de perte des paquets.

Nous présenterons, dans ce qui suit, l'impact de cette variation de la bande passante dans le cas d'un flux de type MP3 et de la voix sur IP en particulier; et ce parce que nous avons observé des résultats similaires que ceux de MP3 pour les flux de type 1AV, 1V, 2AV, 2V et ts.

Les tableaux 6, 7 et 8 illustrent les taux de perte des paquets dans le cas de la variation de la bande passante pour un flux de type VoIP.

trafic généré (Mb/s)	BP (en Mb)	% de perte des paquets
4,5	1,32	0,1
5	0,87	33,44
5,5	0,81	36,24
6	0,73	51,64

Tableau 6. Tableau récapitulatif des taux de perte des paquets en fonction de la BP dans le cas de ROHC

trafic généré (Mb/s)	BP (en Mb)	% de perte des paquets
4	0,98	8,34
4,5	1,06	8,21
5	0,85	13,38
5,5	0,64	81,73

Tableau 7. Tableau récapitulatif des taux de perte des paquets en fonction de la BP dans le cas de deflate

trafic généré (Mb/s)	BP (en Mb)	% de perte des paquets
4	1,98	0,12
4,5	1,46	5,19
5	1,03	10,9
5,5	0,8	19
6	0,66	52,03
6,5	0,52	70,98

Tableau 8. Tableau récapitulatif des taux de perte des paquets en fonction de la BP dans le cas de predictor

Nous remarquons que plus la bande passante est petite, plus le taux de perte est grand.

De même pour les flux vidéo sauf que la taille de la bande passante à partir de laquelle on commence à observer des pertes est plus grande que celle dans le cas de la voix sur IP.

Dans le cas de transfert de fichier, nous n'avons pas noté des pertes de paquets au fur et à mesure que nous limitons la bande passante. En effet, le taux de perte est égal à zéro.

Cependant, cette limitation de la bande passante a un effet sur les temps de transmission et d'interarrivée des paquets qui augmentent énormément.

A partir de notre étude, nous avons pu dégager un ensemble de constatations :

- Le taux de compression est variable suivant la nature des données. En effet, nous avons noté, par exemple, que deflate offre de meilleures performances dans le cas de flux de type Mpeg1 AV. Par contre, cette technique ne donne pas des résultats performants lorsqu'elle s'applique à des données de type MP3.
- L'efficacité de l'algorithme de compression dépend du type des paquets. En effet, le comportement des techniques testées n'est pas le même pour les divers types de flux..
- Le choix de la technique de compression est très important et son efficacité est liée au trafic véhiculé ainsi qu'au taux des erreurs et la bande passante disponible.
- Plus le taux d'erreurs par bits augmente, plus le nombre de paquets perdus augmente. Ceci est attendu vu que le logiciel de réception n'arrive plus à déchiffrer le contenu des paquets et à les lire.
- Le taux de perte des paquets suit une augmentation inversement proportionnelle à la taille de la bande passante. En outre, un réseau chargé influe négativement sur le temps de transmission ainsi que le temps d'interarrivée des paquets et par la suite sur la qualité de réception des données.

8. Conclusions et perspectives

Une étude des travaux réalisés dans ce domaine nous a permis d'orienter notre travail et de ne pas suivre un chemin déjà pris. En effet, plusieurs équipes de recherche ont songé à la sélection, mais ils se sont

basés, dans la plupart des cas, sur l'amélioration au sein d'un même algorithme en choisissant les paramètres de compression convenables.

Nous avons mis en place une plate-forme de mesures permettant de varier différents paramètres de techniques de compression et caractéristiques du réseau afin de voir leur impact.

A partir des résultats que nous avons dégagés, des démarches peuvent être engagées pour concevoir et implémenter l'algorithme de sélection "intelligent" dont nous avons parlé au départ. Nos travaux constituent, dans cette perspective, une première esquisse des critères de sélection de la technique de compression convenable.

Il sera intéressant de poursuivre cette étude en implémentant un algorithme de sélection basé sur ces résultats et ces constatations et en validant cet algorithme après l'avoir intégré dans un réseau téléphonique tel que le GPRS.

Il sera aussi enrichissant d'étudier d'autres critères de sélection et de combiner deux ou plusieurs techniques de compression à savoir une technique de compression de donnée avec une autre qui permet de compresser les en-têtes.

D'autres techniques de compression peuvent faire objet de cette étude telle que la technique de compression d'en-tête EPIC (Efficient Protocol Independent Compression) qui fait l'objet, en ce moment, d'une thèse de doctorat à l'ENST Bretagne. On peut citer aussi la technique de compression d'en-tête TAROC (TCP Aware RObust Header Compression) et la technique de compression de données "bsdcomp".

REFERENCES

- [Car 03] A.Carolina "Compression des en-têtes sur les réseaux bas débit", Thèse de doctorat présenté en vue de l'obtention du titre docteur de l'université Rennes 1, Ecoles Nationale Supérieure des Télécommunications, Décembre 2003.
- [Eth 04] "Site Web du sniffer réseau Ethereal", <http://www.ethereal.com/introduction.html#features>, dernière consultation en Août 2004.
- [Gil 98] F.Gilbert and F.Moutarde."Intelligent compression of still images". *Proceedings of 6th ACM International Multimedia Conference*, Bristol, 12-16 septembre 1998.
- [Gnu 04] "Site Web gnuplot", <http://www.gnuplot.info/>, Dernière consultation en Avril 2004.
- [Smi 95] J.R.Smith et S.Chang, "*Frequency and Spatially Adaptive Wavelet Packets*", *Proceedings of the IEEE, ICASSP'95*, May 1995.
- [Tay 01] C.Taylor et S.Dey, "*Adaptive Image Compression for Enabling Mobile Multimedia Communication*", In *IEEE International Conference on Communication*, June 2001.
- [Wis 01] P.Wischow, "*Lossless Image Compression With Intelligent Algorithm Selection*". Thèse de doctorat, University of South Alabama, May 2001.