Groupe de travail Réseau
Request for Comments : 2507
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

M. Degermark, Lulea University of Technology/SICS
B. Nordgren, Lulea University of Technology/Telia Research AB
S. Pink, Lulea University of Technology/SICS

février 1999

Compression d'en-tête IP

Statut du présent mémoire

La présente RFC spécifie un protocole de normalisation pour la communauté Internet et appelle à des discussions et suggestions pour son amélioration. Prière de se reporter à l'édition en cours des "Internet Official Protocol Standards" (*normes officielles du protocole Internet*) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémo n'est soumise à aucune restriction.

Résumé

Le présent document décrit comment compresser plusieurs en-têtes IP et des en-têtes TCP et UDP par bonds sur des liaisons point à point. Les méthodes peuvent être appliquées aux en-têtes IPv6 de base et d'extension, aux en-têtes IPv4, aux en-têtes TCP et UDP, et aux en-têtes IPv6 et IPv4 encapsulés.

Les en-têtes des paquets UDP ou TCP normaux peuvent être compressés jusqu'à 4-7 octets incluant la somme de contrôle UDP ou TCP de deux octets. Cela retire largement l'impact négatif des grands en-têtes IP et permet une utilisation efficace de la bande passnte sur les liaisons à vitesse faible ou moyenne.

Les algorithmes de compression sont spécifiquement conçus pour bien fonctionner sur des liaisons avec des taux de perte de paquet non triviaux. Plusieurs technologies radio et de modem utilisent de telles liaisons.

Table des Matières

1. Introduction

Il y a plusieurs raisons de faire de la compression d'en-tête sur des liaisons à faible ou moyenne vitesse. La compression d'en-tête peut :

* Améliorer le temps de réponse interactive.
  Pour des liaisons à très basse vitesse, l'écho des caractères peut prendre plus de 100-200 ms à cause du temps nécessaire pour transmettre de gros en-têtes. 100-200 ms est le maximum que les gens peuvent tolérer sans estimer que le système est paresseux.

* Permettre d'utiliser de petits paquets pour des données en vrac avec une bonne efficacité de ligne.
  Ceci est important lorsque du trafic interactif (par exemple, Telnet) et en vrac (par exemple FTP) est mélé parce que les données en vrac devraient être portées dans de petits paquets pour diminuer le temps d'attente lorsque un paquet avec des données interactives est coincé derrière un paquet de données en vrac.

  Utiliser des paquet de petite taille pour le trafic FTP dans ce cas est une solution globale à un problème local. Cela va augmenter la charge du réseau car il doit traiter beaucoup de petits paquets. Une meilleure solution serait de fragmenter localement les gros paquets sur la liaison lente.

* Permettre d'utiliser de petits paquets pour le trafic à faible débit sensible au délai.
  Pour de telles applications, par exemple, la voix, le temps nécessaire pour remplir un paquet avec des données est significatif si les paquets sont grands. Pour avoir un faible délai de bout en bout, les petits paquets sont préférables. Sans compression d'en-tête, les en-têtes IPv6/UDP les plus petits possibles (48 octets) consomment 19,2 kbit/s avec un débit de paquet de 50 paquets/s. 50 paquets/s est équivalent à avoir 20 ms d'échantillons d'équivalent vocal dans chaque paquet. Les en-têtes IPv4/UDP consomment 11,2 kbit/s à 50 paquets/s. Les en-têtes de tunnelage ou d'acheminement, par exemple pour prendre en charge la mobilité, vont augmenter la bande passante consommée par en-tête de 10 à 20 kbit/s. Cela devrait être comparé avec la bande passante exigée par les échantillons sonores réels, par exemple 13 kbit/s avec le codage GSM. La compression d'en-tête peut réduire significativement la bande passante nécessaire pour les en-têtes, d'environ 1,7 kbit/s dans notre exemple. Cela permet une meilleure qualité de transmission de la voix sur les modems à 14,4 et 28,8 kbit/s.

* Diminuer la redondance d'en-tête.
  Une taille courante de segments TCP pour les transferts en vrac sur des liaisons à moyenne vitesse est aujourd'hui de 512 octets. Lorsque les segments TCP sont tunnelés, par exemple à cause de l'utilisation d'IP mobile, les en-têtes IPv4/IPv6/TCP font 100 octets. La compression d'en-tête va diminuer la redondance d'en-tête pour IPv6/TCP de 19,5 pour cent à moins de 1 pour cent, et pour le tunnage IPv4/TCP de 11,7 à moins de 1 pour cent. C'est un gain significatif pour les vitesses de ligne jusqu'à quelques Mbit/s.

  La spécification IPv6 prescrit la découverte de la MTU du chemin, de sorte que les transferts TCP en vrac en IPv6 devraient utiliser des segments supérieurs à 512 octets lorsque c'est possible. Cependant, avec les segments de 1400 octets (l'encapsulation Ethernet de la RFC894 permet des charges utiles de 1500 octets, dont 100 octets sont utilisés pour les en-têtes IP) la compression d'en-tête réduit la redondance d'en-tête IPv6 de 7,1 % à 0,4 %.

* Réduire le taux de perte de paquet sur les liaisons à perte.
  Comme moins de bits sont envoyés par paquet, le taux de perte de paquets sera inférieur pour un taux d'erreurs binaires donné. Il en résulte un débit plus élevé pour TCP car la fenêtre d'envoi peut s'ouvrir plus entre les pertes, et moins de pertes de paquets pour UDP.

Les mécanismes décrits ici sont destinés à une liaison en point à point. Cependant, il faut prendre soin de permettre des extensions pour les liaisons en multi accès et en diffusion groupée.

Les en-têtes qui peuvent être compressés incluent les en-têtes TCP, UDP, IPv4, et IPv6, de base et d'extension. Pour les paquets TCP, les mécanismes de Van Jacobson [RFC1144] sont utilisés pour récupérer des pertes. Deux mécanismes supplémentaires qui augmentent l'efficacité de la compression de Van Jacobson sur les liaisons à perte sont aussi décrits. Pour les paquets non TCP, le démarrage lent de compression et les rafraîchissements périodiques d'en-tête permettent des périodes minimales d'élimination de paquet après la perte d'un en-tête qui change le contexte. Il y a des trucs pour ajouter des schémas de compression d'en-tête par dessus UDP, par exemple, la compression des en-têtes RTP.

La compression d'en-tête s'appuie sur le fait que de nombreux champs sont constants ou changent rarement dans les paquets consécutifs qui appartiennent au même flux de paquets. Les champs qui ne changent pas entre les paquets n'ont pas besoin d'être transmis du tout. Les champs qui changent souvent avec des valeurs petites et/ou prévisibles, par exemple, les numéros de séquence TCP, peuvent être codés de façon incrémentaire de telle sorte que le nombre de bits nécessaire pour ces champs décroisse de façon significative. Seuls les champs qui changent souvent et de façon aléatoire, par exemple, les sommes de contrôle ou les données d'authentification, ont besoin d'être transmises dans chaque en-tête.

Le principe général de la compression d'en-tête est d'envoyer occasionnellement un paquet avec un en-tête complet ; les en-têtes compressés suivants se réfèrent au contexte établi par l'en-tête complet et peuvent contenir des changements incrémentaires du contexte.

Ce schéma de compression d'en-tête n'exige pas que tous les paquets du même flux passent sur la liaison compressée. Cependant, pour les flux TCP, la différence entre les en-têtes successifs peut devenir plus irrégulière et le taux de compression peut diminuer. Pas plus qu'il n'est exigé que les paquets de données TCP et d'accusé de réception correspondants traversent la liaison en directions opposées.

Ce schéma de compression d'en-tête est utile sur les liaisons de premier bond ou de dernier bond aussi bien que sur les liaisons qui sont au milieu du réseau. Lorsque de nombreux flux de paquets (plusieurs centaines) traversent la liaison, peut survenir un phénomène qu'on pourrait appeler le "battage des CID" *(CID thrashing)*, où les en-têtes ne correspondent que rarement à un contexte existant et doivent être envoyés non compressés ou comme en-têtes complets. Il appartient à une mise en œuvre d'utiliser des techniques telles que l'hystérèse pour s'assurer que les flux de paquets qui donnent les plus forts taux de compression conservent leur contexte. De telles techniques seront vraisemblaglement plus nécessaires dans le milieu du réseau.

2. Terminologie

La présente section explique quelques termes utilisés dans le document.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMPANDE", "PEUT", et "FACULTATIF" lorsque ils apparaissent dans ce document, sont à interpréter comme décrit dans la RFC2119.

Sous en-tête
Un en-tête fondé sur IPv6, un en-tête d'extension IPv6, un en-tête IPv4, un en-tête UDP, ou un en-tête TCP.

En-tête
Une chaîne de sous en-têtes.

Compresser
Acte de réduction de la taille d'un en-tête en retirant des champs d'en-tête ou en réduisant la taille de champs d'en-tête. Ceci est fait d'une façon telle qu'un décompresseur puisse reconstruire l'en-tête si son état de contexte est identique à l'état de contexte utilisé lors de la compression de l'en-tête.

Décompresser
Acte de reconstruction d'un en-tête compressé.

Identifiant de contexte (CID)
Petit nombre unique qui identifie le contexte qui devrait être utilisé pour décompresser un en-tête compressé. Il est porté dans les en-têtes complets et dans les en-têtes compressés.

Contexte
État qu'utilise le compresseur pour compresser un en-tête et le décompresseur pour décompresser un en-tête. Le contexte est la version non compressée du dernier en-tête envoyé (au compresseur) ou reçu (au décompresseur) sur la liaison, sauf pour les champs de l'en-tête qui sont inclus "tels quels" dans les en-têtes compressés ou peuvent être déduits, par exemple, la taille de la trame de niveau liaison.

Le contexte pour un flux de paquets est associé à un identifiant de contexte. Le contexte pour les flux de paquets non TCP est aussi associé à une génération.

Génération
Pour un flux de paquets non TCP, chaque nouvelle version du contexte pour un CID donné est associé à une génération : un petit nombre qui est incrémenté chaque fois que le contexte associé à ce CID change. Elle est portée par les en-têtes non TCP complets et compressés.

Flux de paquets
Séquence de paquets dont les en-têtes sont similaires et partagent le contexte. Par exemple, les en-têtes dans un flux de paquets TCP ont la même adresse de source et de destination finale, et les mêmes numéros d'accès dans l'en-tête TCP. De même, les en-têtes dans un flux de paquets UDP ont la même adresse de source et de destination, et les mêmes numéros d'accès dans l'en-tête UDP.

En-tête complet (rafraîchissement d'en-tête)
C'est un en-tête non compressé qui met à jour ou rafraîchit le contexte pour un flux de paquets. Il porte un CID qui sera utilisé pour identifier le contexte.

Les en-têtes complets pour les flux de paquets non TCP portent aussi la génération du contexte qu'ils mettent à jour ou rafraîchissent.

En-tête régulier
C'est un en-tête normal, non compressé. Il ne porte pas de CID ou de génération en association.

Décompression incorrecte
Lorsque un en-tête compressé puis décompressé est différent de l'en-tête non compressé. Cela est habituellement dû à une discordance de contexte entre le compresseur et le décompresseur ou à des erreurs binaires durant la transmission de l'en-tête compressé.

Codage différentiel
C'est une technique de compression dans laquelle la valeur compressée d'un champ d'en-tête est la différence entre la valeur actuelle du champ et la valeur du même champ dans l'en-tête précédent appartenant au même flux de paquets. Un décompresseur peut ainsi obtenir la valeur du champ en ajoutant la valeur qui est dans l'en-tête compressé à son contexte. Cette technique est utilisée pour les flux TCP mais pas pour les flux non TCP.

3. Méthode de compression

Beaucoup des informations de l'en-tête restent les mêmes pendant la durée de vie d'un flux de paquets. Pour les flux de paquets non TCP presque tous les champs des en-têtes sont constants. Pour TCP, de nombreux champs sont constants et d'autres changent avec des valeurs faibles et prévisibles.

Pour initier la compression des en-têtes d'un flux de paquets, un en-tête complet portant un identifiant de contexte, un CID, est transmis sur la liaison. Le compresseur et le décompresseur mémorisent la plupart des champs de cet en-tête complet comme contexte. Le contexte consiste en tous les champs de l'en-tête dont la valeur est constante et n'a donc pas besoin d'être envoyée du tout sur la liaison, ou change peu entre les en-têtes consécutifs de sorte qu'il faut utilisee moins de bits pour envoyer la différence entre la valeur précédente que d'envoyer la valeur absolue.

Tout changement dans les champs dont on s'attend à ce qu'ils soient constants dans un flux de paquets va être cause que le compresseur envoie à nouveau un en-tête complet pour mettre à jour le contexte chez le décompresseur. Tant que le contexte est le même chez le compresseur et le décompresseur, les en-têtes peuvent être décompressés pour qu'ils soient exactement comme ils étaient avant la compression. Cependant, si un en-tête complet ou un en-tête compressé est perdu durant la transmission, le contexte chez le décompresseur peut devenir obsolète car il n'a pas eté mis correctement à jour. Les en-têtes compressés seront alors décompressés de façon incorrecte.

IPv6 n'est pas destiné à être utilisé sur des liaisons qui pourraient livrer une fraction significative de paquets endommagés

au module IPv6. Cela signifie que les liaisons doivent avoir un taux d'erreurs binaires très faible ou que les trames de niveau liaison doivent être protégées par des sommes de contrôle fortes, par la correction d'erreur directe ou quelque chose de cette nature. La compression d'en-tête NE DEVRAIT PAS être utilisée pour IPv4 sans de fortes somme de contrôle de niveau liaison. Les trames endommagées seront donc éliminées par la couche de liaison. La mise en œuvre de couche liaison peut indiquer au module de compression d'en-tête qu'une trame a été endommagée, mais elle ne peut pas dire à quel flux de paquets elle appartenait car ce peut être le CID qui est endommagé. De plus, les trames peuvent disparaître sans que le sache la mise en œuvre de la couche liaison, par exemple, si la liaison est une liaison multi bonds où les trames peuvent être abandonnées à chaque bond à cause de l'encombrement. La sorte d'erreur de liaison qu'un module de compression d'en-tête devrait traiter et contre laquelle il devrait protéger est donc la perte de paquet.

Un schéma de compression d'en-tête a donc besoin de mécanismes pour mettre à jour le contexte chez le décompresseur et pour détecter ou éviter une decompression incorrecte. Ces mécanismes sont très différents pour les flux TCP et non TCP, et sont décrits aux paragraphes 3.2 et 3.3.

Les mécanismes de compression du présent document supposent que les paquets ne sont pas réarrangés entre le compresseur et le décompresseur. Si la liaison réarrange en fait, la section 11 décrit les mécanismes pour ordonner les paquets avant la décompression. On suppose aussi que la mise en œuvre de couche liaison peut fournir la longueur des paquets, et qu'il n'y a pas de bourrage dans les paquets UDP ou les paquets tunnelés.

3.1 Types de paquet

Cette méthode de compression utilise quatre types de paquet en plus des types de paquets IPv4 et IPv6. La combinaison du type de paquet de niveau liaison et de la valeur des quatre premiers bits du paquet détermine de façon univoque le type de paquet. Les détails sur la façon dont ces types de paquet sont représentés figurent à la section 13.

FULL_HEADER – indique un paquet avec un en-tête non compressé, incluant un CID et, si ce n'est pas un paquet TCP, une génération. Il établit ou rafraîchit le contexte pour le flux de paquets identifié par le CID.

COMPRESSED_NON_TCP – indique un paquet non TCP avec un en-tête compressé. L'en-tête compressé consiste en un CID qui identifie quel contexte utiliser pour la décompression, une génération pour détecter un contexte incohérent et les champs de l'en-tête qui changent de façon aléatoire.

COMPRESSED_TCP – indique un paquet avec un en-tête TCP compressé, contenant un CID, un octet fanion qui identifie quels champs ont changé, et les champs changés codés comme différence par rapport à la valeur précédente.

COMPRESSED_TCP_NODELTA – indique un paquet avec un en-tête TCP compressé où tous les champs qui sont normalement envoyés comme différence à la valeur précédente sont envoyés en l'état. Ce type de paquet n'est envoyé qu'en réponse à une demande d'en-tête provenant du décompresseur. Il ne doit pas être envoyé suite à une retransmission.

En plus des types de paquet utilisés pour la compression, les paquets IPv4 et IPv6 réguliers sont utilisés chaque fois que la compresseur décide de ne pas compresser un paquet. Un type de paquet supplémentaure peut être utilisé pour accélérer la réparation d'un flux TCP sur des liaisons où le décompresseur peut envoyer des paquets au compresseur.

CONTEXT_STATE – indique un paquet spécial envoyé du décompresseur au compresseur pour communiquer une liste de CID (TCP) pour lesquels la synchronisation a été perdue. Ce paquet n'est envoyé que sur une seule liaison de sorte qu'il n'exige pas d'en-tête IP. Son format figure au paragraphe 10.2.

3.2 Paquets perdus dans les flux de paquets TCP

Comme les en-têtes TCP sont compressés en utilisant la différence avec l'en-tête TCP précédent, la perte d'un paquet avec un en-tête compressé ou complet va être cause que les en-têtes compressés suivants seront décompressés incorrectement parce que le contexte utilisé pour la décompression n'aura pas été incrémenté de la façon appropriée.

La perte d'un en-tête TCP compressé va avoir pour conséquence que les numéros de séquence des en-têtes TCP décompressés ultérieurement vont être faux de k, où k est la taille du segment perdu. De tels en-têtes TCP incorrectement décompressés seront éliminés par le receveur TCP car la somme de contrôle TCP capture de façon fiable les erreurs "faux de k" dans les numéros de séquence pour un k plausible.

Les mécanismes de réparation de TCP vont finalement retransmettre le segment éliminé et le compresseur surveille les en-têtes TCP pour détecter quand TCP retransmet. Quand cela arrive, le compresseur envoie un en-tête complet en supposant que la retransmission était due à une discordance d'état de compression chez le décompresseur. La [RFC1144] donne une

bomme explication de ce mécanisme.

Les mécanismes de la section 10 devraient être utilisés pour accélérer la réparation du contexte. C'est important sur les liaisons à moyenne vitesse avec de forts taux de perte de paquet, par exemple par radio. Perdre les paquets écoulés pendant une temporisation du fait d'une incohérence de contexte après chaque perte de paquet sur la liaison n'est pas acceptable, tout spécialement lorsque la connexion TCP est sur une large zone.

3.3 Paquets perdus dans les flux de paquets UDP et autres flux non TCP

Les en-têtes incorrectement décompressés de paquets UDP et autres paquets non TCP ne sont pas aussi bien protégés par les sommes de contrôle que les paquets TCP. Il n'y a pas de numéros de séquence qui deviennent "faux de k" et qui garantissent virtuellement un échec de somme de contrôle comme pour TCP. La somme de contrôle UDP ne couvre que la charge utile, l'en-tête UDP, et un pseudo en-tête. Le pseudo en-tête comporte les adresses de source et de destination, le type de protocole de transport et la longueur du paquet de transport. Sauf pour ces champs, de larges parties de l'en-tête IPv6 ne sont pas couvertes par la somme de contrôle UDP. De plus, les autres en-têtes non TCP manquent aussi de sommes de contrôle, par exemple, les fragments.

Afin d'éviter en toute sécurité une décompression incorrecte des en-têtes non TCP, chaque version du contexte pour les flux de paquets non TCP est identifiée par une génération, un petit nombre qui est porté par les en-têtes complets qui établissent et rafraîchissent le contexte. Les en-têtes compressés portent la valeur de la génération du contexte qui était utilisé pour les compresser. Lorsque un décompresseur voit qu'un en-tête compressé porte une valeur de génération autre que celle de la génération de son contexte pour ce flux de paquets, le contexte n'est pas à jour et le paquet doit être éliminé ou mémorisé jusqu'à ce qu'un en-tête complet établisse le contexte correct.

Le codage différentiel n'est pas utilisé pour les flux non TCP, de sorte que les en-têtes compressés non TCP ne changent pas le contexte. Donc, la perte d'un en-tête compressé n'invalide pas les paquets suivants avec des en-têtes compressés. De plus, la génération ne change que lorsque le contexte d'un en-tête complet est différent du contexte de l'en-tête complet précédent. Cela signifie que la perte d'un en-tête complet ne va rendre obsolète le contexte du décompresseur que lorsque l'en-tête complet changerait en fait le contexte.

Le champ Génération est long de 6 bits de sorte que la valeur de la génération se répète après 64 changements de contexte. Pour éviter une décompression incorrecte après des salves d'erreurs ou autres perturbations temporaires, le compresseur ne doit pas réutiliser la même valeur de génération après un délai plus court que MIN_WRAP secondes. Un décompresseur qui a été déconnecté pendant MIN_WRAP secondes ou plus doit attendre le prochain en-tête complet avant de décompresser. Un compresseur doit attendre au moins MIN_WRAP secondes après l'amorçage avant de compresser des en-têtes non TCP. Au lieu de réutiliser trop tôt une valeur de génération , un compresseur peut passer à un autre CID ou envoyer des en-têtes réguliers jusqu'à ce que MIN_WRAP secondes se soient écoulées. La valeur de MIN_WRAP est à la section 14.

3.3.1 Démarrage lent de compression

Pour permettre au décompresseur de récupérer rapidement de la perte d'un en-tête complet qui aurait changé le contexte, les en-têtes complets sont envoyés périodiquement avec une période exponentiellement croissante après un changement du contexte. Cette technique évite un échange de messages entre compresseur et décompresseur utilisé par les autres schémas de compression, tels que dans la [RFC1553]. Un tel échange peut être coûteux pour les mobiles sans fil car plus de puissance est consommée par l'émetteur et des délais peuvent être introduits par le passage entre l'envoi et la réception. De plus, les techniques qui requièrent un échange de messages ne peuvent pas être utilisées sur des liaisons simplex, telles que des canaux par satellite à diffusion directe ou des systèmes de télévision par câble, et sont difficiles à adapter à la diffusion groupée sur des liaisons multi accès.

```
| . | . . | . . . . | . . . . . . . . | . . . . . . . . . . . . . . . | . . . . . . . . . . . . . . . . . . . . . . . . . . . .
↑
Changement.   Paquets envoyés : |   avec en-tête complet,          . avec en-tête compressé
```

Le dessin montre comment les paquets sont envoyés après un changement. Le compresseur conserve une variable pour chaque flux de paquets non TCP, F_PERIOD, qui garde trace de la façon dont de nombreux en-têtes compressés peuvent être envoyés entre les en-tête complets. Lorsque les en-têtes d'un flux de paquets non-TCP changent de telle sorte que son contexte change, un en-tête complet est envoyé et F_PERIOD est réglé à un. Après l'envoi de F_PERIOD en-têtes compressés, un en-tête complet est envoyé. F_PERIOD est doublé chaque fois qu'un en-tête complet est envoyé durant le démarrage lent de compression.

3.3.2 Rafraîchissement périodique d'en-tête

Pour éviter de perdre trop de paquets si un receveur a perdu son contexte, il y a une limite supérieure, F_MAX_PERIOD, au nombre de paquets non TCP avec des en-têtes compressés qui peuvent être envoyés entre les rafraîchissements d'en-tête. Si un paquet est sur le point d'être envoyé et si F_MAX_PERIOD en-têtes compressés ont été envoyés depuis l'envoi dn dernier en-tête complet pour ce flux de paquets, un en-tête complet doit être envoyé.

Pour éviter de longues périodes de déconnexion pour les flux de paquets à faible débit de données, il y a aussi une limite supérieure, F_MAX_TIME, au temps écoulé entre en-têtes complets dans un flux de paquets non-TCP. Si un paquet est à envoyer et si plus de F_MAX_TIME secondes se sont écoulées depuis l'envoi du dernier en-tête complet pour ce flux de paquets, un en-tête complet doit être envoyé. Les valeurs de F_MAX_PERIOD et de F_MAX_TIME figurent à la section 14.

3.3.3 Règles d'envoi des en-têtes complets

Le compresseur peut utiliser le pseudo code suivant pour déterminer quand envoyer un en-tête complet pour un flux de paquets non TCP. Le code utilise deux variables :
C_NUM -- compte du nombre d'en-têtes compressés envoyés depuis l'envoi du dernier en-tête complet
F_LAST -- heure de l'envoi du dernier en-tête complet

et utilise les fonctions
current_time() retourne l'heure actuelle
min(a,b) retourne le plus petit de a et b

Les procédures send_full_header() *(envoyer l'en-tête complet)*, increment_generation_value() *(incrémenter la valeur de génération)*, et send_compressed_header() *(envoyer l'en-tête compressé)* remplissent les fonctions désignées par leur nom.

```
si ( <cet en-tête change le contexte> )
        C_NUM := 0;
        F_LAST := current_time();
        F_PERIOD := 1;
        increment_generation_value();
        send_full_header();
autrement si ( C_NUM >= F_PERIOD )
        C_NUM := 0;
        F_LAST := current_time();
        F_PERIOD := min(2 * F_PERIOD, F_MAX_PERIOD);
        send_full_header();
autrement si ( current_time() > F_LAST + F_MAX_TIME )
        C_NUM := 0;
        F_LAST := current_time();
        send_full_header();
autrement
        C_NUM := C_NUM + 1
        send_compressed_header();
fin du si
```

3.3.4 Coût d'envoi des rafraîchissements d'en-tête

Si chaque f^e paquet porte un en-tête complet, H est la taille d'un en-tête complet, et C est la taille d'un en-tête compressé, la taille moyenne d'un en-tête est (H-C)/f + C.

Pour f > 1, la taille moyenne d'en-tête est de (H - C)/f plus grande qu'un en-tête compressé.

Dans un diagramme où la taille moyenne d'en-tête est donnée pour diverses valeurs, il y a des infléchissements distincts dans la courbe, c'est-à-dire qu'il y a une limite au delà de laquelle l'accroissement de f donne une diminution de la réponse. F_MAX_PERIOD devrait être choisi comme une fréquence se situant bien à droite de l'infléchissement de la courbe. Pour les tailles normales de H et de C, disons 48 octets pour l'en-tête complet (IPv6/UDP) et 4 octets pour l'en-tête compressé, régler F_MAX_PERIOD > 44 signifie que les en-têtes complets vont contribuer pour moins d'un octet à la taille moyenne d'en-tête. Avec un en-tête d'acheminement de quatre adresses, F_MAX_PERIOD > 115 aura le même effet.

La valeur de F_MAX_PERIOD par défaut de 256 (section 14) met la fréquence de l'en-tête complet bien à droite de l'infléchissement et signifie que les en-têtes complets vont normalement contribuer de beaucoup moins qu'un octet à la

taille moyenne d'en-tête. Pour H = 48 et C = 4, les en-têtes complets contribuent d'environ 1,4 bits à la taille moyenne d 'en-tête après avoir atteint en régime peremanent la fréquence de rafraîchissement d'en-tête déterminée par le F_MAX_PERIOD par défaut. 1,4 bits est une redondance très faible.

Après un changement de contexte, le schéma de retard exponentiel va initialement envoyer fréquemment des en-têtes complets. La F_MAX_PERIOD par défaut sera atteinte après l'envoi de neuf en-têtes complets et 255 en-têtes compressés. Ceci est équivalent à un peu plus de 5 secondes pour un flux vocal typique avec des échantillons de voix de 20 ms par paquet.

Durant toute la période d'attente, les en-têtes complets contribuent pour 1,5 octets à la taille moyenne d'en-tête lorsque H = 48 et C = 4. Pour des échantillons de voix de 20 ms, il faut moins de 1,3 secondes pour que les en-têtes complets contribuent pour moins d'un octet à la taille moyenne d'en-tête, et durant ces 1,3 secondes initiales, les en-têtes complets ajoutent moins de 4 octets à la taille moyenne d'en-tête. Le coût du retard exponentiel n'est pas grand et comme les en-têtes des flux de paquets non TCP sont supposés ne changer que rarement, il sera amorti sur une longue durée.

Le coût des rafraîchissements d'en-tête en termes de bande passante est plus élevé que les coûts similaires pour les schémas à état fixe comme celui de la [RFC1553] où les en-têtes complets doivent être acquittés par le décompresseur avant que les en-têtes compressés puissent être envoyés. De tels schémas envoient normalement un en-tête complet plus quelques messages de contrôle lorsque le contexte change. Les schémas à état fixe requièrent plus de types de messages de protocole et un échange de messages est nécessaire. Les schémas à état fixe doivent aussi traiter explicitement des diverses conditions d'erreur que les états conditionnels traitent automatiqquement, par exemple, le cas d'une partie qui disparaît de façon inattendue, situation courante sur les liaisonss radioélectriques où les mobiles peuvent sortir de la portée d'une station de base.

L'avantage majeur du schéma d'état conditionnel est qu'aucune prise de contact n'est nécessaire entre le compresseur et le décompresseur, de sorte que le schéma peut être utilisé sur des liaisons simplex. Les coûts en termes de bande passante sont plus élevés que pour les schémas à état fixe, mais la simplicité du décompresseur, la simplicité du protocole, et l'absence de prise de contact entre compresseur et décompresseur justifient ce petit coût. De plus, les schémas d'état conditionnel sont plus facilement étendus à la diffusion groupée sur des liaisons en multi accès, par exemple des liaisons radio électriques.

4. Groupage des paquets en flux de paquets

La présente section explique comment les paquets PEUVENT être groupés en flux de paquets pour la compression. Pour réaliser les meilleurs taux de compression, les paquets DEVRAIENT être groupés ensemble de telle sorte que les paquets dans le même flux de paquets aient des en-têtes similaires. Si ce groupement échoue, les performances de compression d'en-tête seront mauvaises, car l'algorithme de compression peut rarement utiliser le contexte existant pour le flux de paquets et les en-têtes complets doivent fréquemment être utilisés.

Le groupement est fait par le compresseur. Un compresseur peut utiliser tout critère qu'il estime approprié pour grouper les paquets en flux de paquets. Pour déterminer à quel flux de paquets appartient un paquet, un compresseur PEUT

a) examiner la chaîne compressible de sous en-têtes (voir la section 7),

b) examiner le contenu d'un en-tête de protocole de couche supérieure qui suit la chaîne compressible de sous en-têtes, par exemple des en-têtes ICMP, DVMRP, ou des en-têtes IPX tunnelés,

c) utiliser des informations obtenues d'un gestionnaire de ressource, par exemple si un gestionnaire de ressource demande la compression pour un flux de paquets particulier et fournit un moyen d'identifier les paquets qui appartiennent à ce flux de paquets,

d) utiliser toutes autres informations pertinentes, par exemple si des chemins échouent et si le champ de limite de bonds (TTL) dans un flux de paquets change fréquemment entre n et n+k, un compresseur peut choisir de grouper les paquets dans deux flux de paquets différents.

Un compresseur a toute liberté pour ne pas grouper les paquets dans des flux de paquets pour la compression, en laissant quelques paquets conserver leurs en-têtes normaux et en les passant sans modification.

Tant qu'on suit les règles sur le moment où envoyer les en-têtes complets pour un flux de paquets non-TCP et que les sous en-têtes sont compressés comme spécifié dans le présent document, le décompresseur est capable de reconstruire un en-tête compressé correctement sans considération de la façon dont les paquets sont groupés dans le flux de paquets.

4.1 Lignes directrices pour le groupage des paquets

Ce paragraphe donne des lignes directrices FACULTATIVES sur la façon dont un compresseur peut grouper les paquets en flux de paquets pour la compression.

Champs de définition
Les champs de définition d'un en-tête devraient être présents et identiques dans tous les paquets qui appartiennent au même flux de paquets. Ces champs sont marqués DEF à la section 7. Les champs de définition incluent l'étiquette de flux, les adresses de source et de destination des en-têtes IP, l'adresse de destination finale dans les en-têtes d'acheminement, les prochains champs d'en-tête (pour IPv6), le champ protocole (IPv4), les numéros d'accès (UDP et TCP), et le SPI dans les en-têtes d'authentification et de chiffrement.

Paquets fragmentés
Les paquets fragmentés et non fragmentés ne devraient jamais être groupés dans le même flux de paquets. Le champ Identification de l'en-tête de fragment ou de l'en-tête IPv4 ne devrait pas être utilisé pour identifier le flux de paquets. Si il l'était, le premier fragment d'un nouveau paquet causerait un démarrage lent de compression.

Aucun champ après un en-tête de fragment, ou un en-tête IPv4 pour un fragment, ne devrait être utilisé à des fins de groupement.

Identification de protocole supérieur
Le premier prochain champ d'en-tête identifiant un en-tête non décrit à la section 7 devrait être utilisé pour identifier le flux de paquets, c'est-à-dire, tous les paquets avec les mêmes champs DEF et le même protocole supérieur devraient être groupés.

Champ TTL (champ Limite de bonds)
Une mise en œuvre sophistiquée peut surveiller le champ TTL (Limite de bonds) et si il change fréquemmennt, l'utiliser comme un champ DEF. Cela peut arriver lorsque de fréquentes oscillations de route font que les paquets traversent des chemins différents à travers l'internet.

Champ Classe de trafic (IPv6), champ Type de service (IPv4)
Il est possible que le champ Classe de trafic de l'en-tête IPv6 et le champ Type de service de l'en-tête IPv4 changent fréquemment entre des paquets qui ont par ailleurs des champs DEF identiques. Une mise en œuvre sophistiquée devrait surveiller cela et être prête à utiliser ces champs comme champs de définition.

Lorsque des paquets IP sont tunnelés, ils sont encapsulés avec un en-tête IP supplémentaire au point d'entrée du tunnel et envoyés ensuite au point d'extrémité du tunnel. Pour grouper de tels paquets en un flux de paquets, les en-têtes internes devraient aussi être examinés pour déterminer le flux de paquets. Si ce n'est pas fait, les en-têtes complets seront envoyés chaque fois que changent les en-têtes du paquet IP interne. Aussi lorsque un paquet est tunnelé, les champs d'identification des sous en-têtes internes devraient être pris en considération en plus des champs d'identification de l'en-tête IP initial.

Une mise en œuvre peut utiliser pour l'identification d'autres champs que ceux décrits ici. Si trop de champs sont utilisés pour l'identification, les performances peuvent en souffrir parce que plus de CID seront utilisés et les mauvais CID peuvent être réutilisés lorsque de nouveaux flux ont besoin de CID. Si trop peu de champs sont utilisés pour l'identification, les performances peuvent en souffrir parce qu'il y aura des changements de contexte trop fréquents.

On souligne que ces lignes directrices sont de simple souhaits. Lorsque IPv6 sera largement déployeé et que le trafic IPv6 pourra être analysé, on pourra trouver que d'autres algorithmes de groupement ont de meilleures performances. On souligne également que si le groupement échoue, il en résultera de mauvaises performances mais pas une décompression incorrecte. La décompresseur peut accomplir sa tâche sans considération de la façon dont fonctionne l'algorithme de groupement.

5. Questions de taille

5.1 Identifiants de contexte

Les identifiants de contexte peuvent être longs de 8 ou de 16 bits. Leur taille ne sert pas à trouver le contexte. Un CID de 8 bits avec une valeur de deux et un CID de 16 bits avec une valeur de deux sont équivalents.

Les espaces de CID pour TCP et non TCP sont distincts, de sorte qu'un CID TCP et un CID non TCP n'identifient jamais le même contexte. Même si ils ont la même valeur. Cela double l'espace de CID disponible tout en utilisant le même nombre de bits pour les CID. Il est toujours possible de dire si un en-tête complet ou compressé est pour un paquet TCP ou non TCP, et donc aucun mélange ne peut survenir.

Les en-têtes compressés non TCP codent la taille du CID en utilisant un bit dans le second octet de l'en-tête compressé. Le CID de 8 bits permet une taille minimum d'en-tête compressé de 2 octets pour les paquets non TCP, le CID utilise le premier octet et le bit de taille et la valeur de génération de 6 bits dans le second octet.

Pour TCP, la seule taille de CID disponible est de 8 bits comme dans la [RFC1144]. 8 bits est probablement suffisant car les connexions TCP sont toujours en point à point.

La taille de CID de 16 bits peut n'être pas nécessaire pour les liaisons point à point ; il est prévu de l'utiliser sur des liaisons multi-accès où un plus grand espace de CID pourrait être nécessaire pour l'efficacité du choix des CID.

La difficulté majeure avec les liaisons multi-accès est que plusieurs compresseurs partagent l'espace de CID d'un décompresseur. Les CID ne peuvent plus être choisis indépendamment par les compresseurs car des collisions peuvent survenir. Ce problème peut se résoudre en laissant les décompresseurs avoir un espace de CID distinct pour chaque compresseur. Avoir des espaces de CID distincts exige que les décompresseurs puissent identifier le compresseur qui envoie le paquet compressé, peut-être en utilisant des informations de couche liaison comme à qui envoyer la trame de couche liaison. Si de telles informations ne sont pas disponibles, tous les compresseurs sur la liaison multi-accès peuvent être énumérée, automatiquement ou autrement, et fournir leur numéro au titre du CID. Cette dernière méthode exige un grand espace de CID.

5.2 Taille du contexte

La taille du contexte DEVRAIT être limitée pour simplifier la mise en œuvre du compresseur et du décompresseur, et mettre une limite à leurs exigences de mémoire. Cependant, il n'y a pas de limite supérieure à la taille d'un en-tête IPv6 car la chaîne des en-têtes d'extension peut être d'une longueur arbitraire. C'est un problème car le contexte est essentiellement un en-tête mémorisé.

Le paramètre configurable MAX_HEADER (voir à la section 14) représente la taille maximum du contexte, exprimée comme l'en-tête de taille maximum qui peut être mémorisé comme contexte. Lorsque un en-tête est plus grand que MAX_HEADER, seule une partie en est mémorisée comme contexte. Une mise en œuvre NE DOIT PAS compresser plus que les MAX_HEADER octets initiaux d'un en-tête. Une mise en œuvre NE DOIT PAS comprimer partiellement un sous en-tête.

Donc, la partie de l'en-tête qui est mémorisée comme contexte et est compressée est la plus longue séquence initiale des sous en-têtes entiers qui n'est pas supérieure à MAX_HEADER octets.

5.3 Taille des en-têtes complets

Il est souhaitable d'éviter d'augmenter la taille des paquets avec des en-têtes complets au delà de leur taille d'origine, car leur taille peut être optimisée pour la MTU de la liaison. Comme on suppose que la mise en œuvre de la couche liaison fournit la longueur des paquets, on peut utiliser les champs de longueur dans les en-têtes complets pour passer les valeurs du CID et de génération au décompresseur.

Cela exige que la couche liaison n'ajoute pas de bourrage à la charge utile, et au moins de ne pas effectuer de bourrage sur ce qui peut être délivré à l'usager de la liaison de destination. Il est aussi exigé qu'aucun bourrage suppélmentaire ne soit ajouté après les données UDP ou dans les paquets tunnelés. Cela permet que les valeurs des champs de longueur soient calculées à partir de la longueur des en-têtes et de la longueur de la trame de couche liaison.

La génération requiert un octet et le CID peut exiger jusqu'à 2 octets. Il y a des champs de longueur de 2 octets dans l'en-tête de base IPv6, l'en-tête IPv4, et l'en-tête UDP.

Un en-tête TCP complet va donc avoir au moins deux octets disponibles dans l'en-tête IP pour passer les 8 bits du CID, ce qui est suffisant. Il y aura plus de deux octets disponibles si il y a plus d'un en-tête IP.

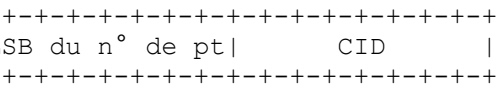
La [RFC1144] utilise le champ Protocole de huit bits de l'en-tête IPv4 pour passer le CID. On ne peut pas utiliser la méthode correspondante car la séquence d'en-tête d'extension IPv6 n'est pas fixée et les valeurs de CID ne sont pas disjointes des valeurs légales des champs Prochain en-tête.

Un paquet IPv6/UDP ou IPv4/UDP va avoir quatre octets disponibles pour passer la génération et le CID, de sorte que toutes les tailles de CID peuvent être utilisées. Les flux de paquets fragmentés ou chiffrés ne peuvent avoir que deux octets disponibles pour passer la génération et le CID. Donc, les CID de 8 bits peuvent être la seule taille de CID qui puisse être utilisée pour de tels flux de paquets. Lorsque le tunnelage IPv6/IPv4 ou IPv4/IPv6 est utilié, il y aura au moins quatre octets disponibles, et les deux tailles de CID peuvent être utilisées.

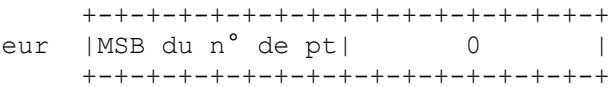
La valeur de génération est passée dans l'octet de poids fort du premier champ de longueur dans l'en-tête complet. Lorsque un seul champ de longueur est disponible, le CID de huit bits est passé dans l'octet de moindre poids. Lorsque deux champs de longueur sont disponibles, les deux plus faibles octets du CID sont passés dans le second champ de longueur et l'octet de moindre poids du premier champ de longueur porte le plus fort octet du CID.

5.3.1 Utilisation des champs de longueur dans les en-têtes TCP complets

Utiliser le premier champ de longueur :



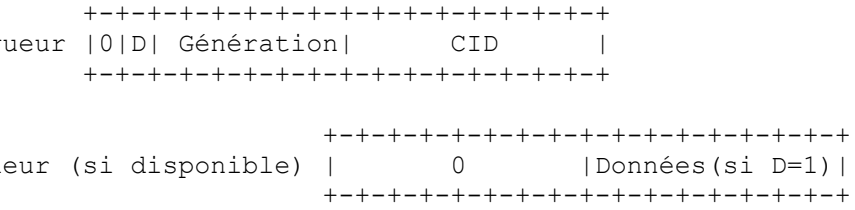
Utiliser le second champ de longueur si disponible :



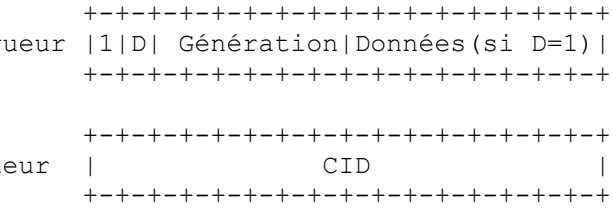
"n° de pt est une abréviation pour "numéro de séquence du paquet, décrit au paragraphe 11.2.

5.3.2 Utilisation des champs de longueur dans les en-têtes non TCP complets

En-têtes non TCP complets avec CID de huit bits :



En-têtes non TCP complets avec CID de 16 bitss :

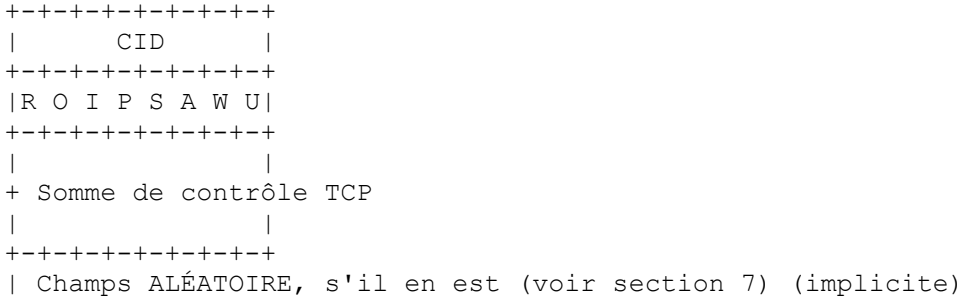


Le premier bit dans le premier champ Longueur indique la longueur du CID. Le champ Données est à zéro si D est zéro. L'utilisation du bit D et du champ Données est expliquée à la section 12.

6. Formats d'en-tête compressé

La présente section utilise des termes (DELTA, ALÉATOIRE) définis à la section 7.

a) Format COMPRESSED_TCP (similaire à celui de la [RFC1144]) :



```
 - - - - - - - -
| R-octet       |                        (si R=1)
 - - - - - - - -
| Valeur du pointeur Urgent              (si U=1)
 - - - - - - - -
| Delta de fenêtre                       (si W=1)
 - - - - - - - -
| Delta du numéro d'accusé de réception (si A=1)
 - - - - - - - -
| Delta du numéro de séquence            (si S=1)
 - - - - - - - -
| Delta d'identification IPv4            (si I=1)
 - - - - - - - -
|  Options                               (si O=1)
 - - - - - - - -
```

Les derniers fanions dans le second octet (IPSAWU) ont la même signification que dans la [RFC1144], sans considérer si les segments TCP sont portés par IPv6 ou IPv4. Le bit C a été éliminé parce que le CID est toujours présent. Le contexte associé au CID garde trace de la version IP et des champs ALÉATOIRE présents. L'ordre entre les champs delta spécifiés ici est exactement celui de la [RFC1144]. Une mise en œuvre va normalement examiner le contexte depuis le début et insérer les champs ALÉATOIRE dans l'ordre. Les champs ALÉATOIRE sont donc placés avant les champs DELTA de l'en-tête TCP dans le même ordre que celui dans lequel ils surviennent dans l'en-tête non compressé d'origine.

Le fanion I est à zéro sauf si un en-tête IPv4 précède immédiatement l'en-tête TCP. L'en-tête combiné IPv4/TCP est alors compressé comme une unité comme décrit dans la [RFC1144]. Les champs Identification dans les en-têtes IPv4 qui ne sont pas immédiatement suivis par un en-tête TCP sont ALÉATOIRE.

Si le fanion O est mis, les options de l'en-tête TCP n'étaient pas les mêmes que dans l'en-tête précédent. Le champ Option entier est placé en dernier dans l'en-tête TCP compressé.

Si le fanion R est mis, il y a des différences entre le contexte et le champ Réservé (6 bits) dans l'en-tête TCP ou les bits 6 ou 7 de l'octet TOS (octet Classe de trafic) dans un en-tête IPv4 (en-tête IPv6) qui précède immédiatement l'en-tête TCP. Un octet avec les valeurs réelles du champ Réservé et les bits 6 et 7 du champ TOS ou Classe de trafic est alors placé immédiatement après les champs ALÉATOIRE. Les bits 0 à 5 de l'octet passé sont la valeur réelle du champ Réservé, et les bits 6 et 7 sont les valeurs réelles des bits 6 et 7 dans le champ TOS ou Classe de trafic. Si il n'y a pas d'en-tête IP qui précède, les bits 6 et 7 sont 0. L'octet passé avec le fanion R NE DOIT PAS mettre à jour le contexte.

Note : L'octet R ne met pas à jour le contexte parce que si il le faisait, la somme de contrôle nTCP ne protègerait pas le TCP receveur contre la décompression erronée d'en-têtes. Les bits 6 et 7 de l'octet TOS ou Classe de trafic sont supposés changer fréquemment du fait de la Notification explicite d'encombrement.

Voir au paragraphe 7.12 et la [RFC1144] des informations complémentaires sur comment compresser les en-têtes TCP.

b) Format d'en-tête COMPRESSED_TCP_NODELTA :

```
+-+-+-+-+-+-+-+-+
|      CID      |
+-+-+-+-+-+-+-+-+
|  Champs ALÉATOIRE, s'il en est (voir section 7) (implicite)
+-+-+-+-+-+-+-+-+
|  En-tête TCP complet excepté les numéros d'accès
+-+-+-+-+-+-+-+-+
```

c) En-tête compressé nonTCP, CID de 8 bits :

```
 0             7
+-+-+-+-+-+-+-+-+
|      CID      |
+-+-+-+-+-+-+-+-+
|0|D| Génération|
+-+-+-+-+-+-+-+-+
|      données  |                      (si D=1)
 - - - - - - - -
| Champs ALÉATOIRE, s'il en est (voir section 7) (implicite)
```

```
 - - - - - - - -
```

d) En-tête compressé nonTCP, CID de 16 bits :

```
 0             7
+-+-+-+-+-+-+-+-+
|  MSB du CID   |
+-+-+-+-+-+-+-+-+
|1|D| Génération|
+-+-+-+-+-+-+-+-+
|  LDB du CID   |
+-+-+-+-+-+-+-+-+
|      données  |                       (si D=1)
 - - - - - - - -
| Champs ALÉATOIRE, s'il en est (voir section 7) (implicite)
 - - - - - - - -
```

La génération, le CID et les données facultatives de un octet sont suivis par les champs ALÉATOIRE pertinents (voir la section 7) comme impliqué par l'état de compression, placés dans le même ordre que celui dans lequel ils surviennent dans l'en-tête non compressé d'origine, suivis par la charge utile.

7. Compression des sous en-têtes

La présente section donne les règles sur la façon de compresser les chaînes de sous en-têtes compressibles. Ces règles DOIVENT être suivies. Les sous en-têtes qui peuvent être compressés incluent les en-têtes IPv6 de base et d'extension, les en-têtes TCP, les en-têtes UDP, et les en-têtes IPv4. La chaîne de sous en-têtes compressible s'étend depuis le début de l'en-tête

a) jusqu'au premier, sans l'inclure, en-tête qui n'est pas un en-tête IPv4, un en-tête IPv6 de base ou d'extension, un en-tête TCP, ou un en-tête UDP, ou
b) jusqu'au, et inclus, premier en-tête TCP, en-tête UDP, en-tête de fragment, en-tête d'encapsulation de charge utile de sécurité, ou en-tête IPv4 pour un fragment, quelque soit celui qui donne la plus courte chaîne. Par exemple, les règles a) et b) conviennent toutes deux pour une chaîne de sous en-têtes qui contient un en-tête de fragment et se termine à un paquet IPX tunnelé. Comme la règle b) donne une plus sourte chaîne, la chaîne compressible de sous en-têtes s'arrête à l'en-tête de fragment.

Les paragraphes qui suivent présentent une classification systématique de la façon dont il est attendu que changent tous les champs des sous en-têtes.

NOCHANGE Le champ ne devrait pas changer. Tout changement signifie qu'un en-tête complet DOIT être envoyé pour mettre le contexte à jour.

DELTA Le champ peut changer souvent mais habituellement, la différence avec le champ de l'en-tête précédent est petite, de sorte qu'il est meilleur marché d'envoyer le changement par rapport à la valeur précédente que la valeur actuelle. Ce type de compression n'est utilisé que pour les flux de paquets TCP.

ALÉATOIRE Le champ doit être inclus "tel quel" dans les en-têtes compressés, normalement parce que les changements sont inprévisibles.

INFÉRÉ Le champ contient une valeur qui peut être inférée d'autres valeurs, par exemple de la taille de la trame qui porte le paquet, et ne doit donc pas être inclus dans l'en-tête compressé.

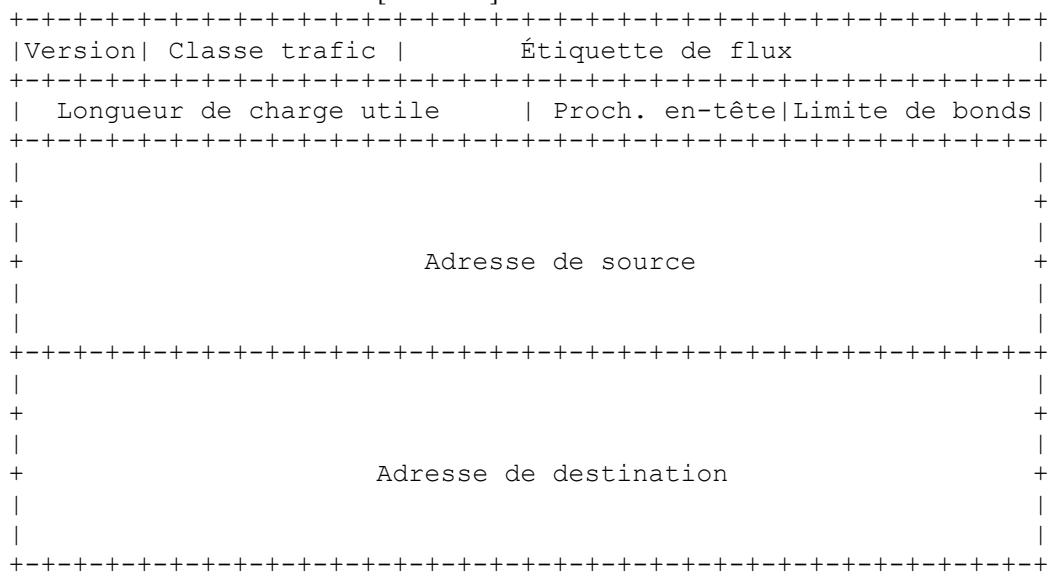
La classification implique comment est construit l'en-tête compressé. Aucun champ qui est NOCHANGE ou INFÉRÉ n'est présent dans un en-tête compressé. Un compresseur obtient les valeurs des champs NOCHANGE du contexte identifié par l'identifiant de compression, et obtient les valeurs de INFÉRÉ de la mise en œuvre de la couche liaison, par exemple, de la taille de la trame de couche liaison, ou d'autres champs, par exemple, en recalculant la somme de contrôle d'en-tête IPv4. Les champs DELTA sont codés comme la différence avec la valeur dans le paquet précédent dans le même flux de paquets. Le décompresseur doit mettre à jour le contexte en ajoutant la valeur dans l'en-tête compressé à la valeur dans son contexte. Le résultat est la propre valeur du champ. Les champs ALÉATOIRE doivent être envoyés "tels quels" dans l'en-tête compressé. Les champs ALÉATOIRE doivent survenir dans l'en-tête compressé dans le même ordre que celui qu'ils occupent dans l'en-tête complet.

Les champs qui peuvent être facultativement utilisés pour identifier à quel flux de paquets appartient un paquet

conformément au paragraphe 4.1 sont marqués avec le mot DEF. Pour un compresseur qui utilise les lignes directrices facultatives du paragraphe 4.1, toute différence dans les champs DEF correspondants entre deux paquets implique qu'ils appartiennent à des flux de paquets différents. De plus, si un champ DEF est présent dans un paquet mais pas dans un autre, les paquets appartiennent à des flux de paquets différents.

7.1 En-tête IPv6

L'en-tête IPv6 est décrit à la section 3 de la [RFC2460].



Version	NOCHANGE (DEF)
Classe de trafic	NOCHANGE (peut être DEF, voir § 4.1) (voir aussi le § 6 a)
Étiquette de flux	NOCHANGE (DEF)
Longueur de charge utile	INFÉRÉ
Prochain bond	NOCHANGE
Limite de bonds	NOCHANGE (peut être DEF, voir § 4.1)
Adresse de source	NOCHANGE (DEF)
Adresse de destination	NOCHANGE (DEF)

Le champ Longueur de charge utile des en-têtes encapsulés doit correspondre à la valeur de longueur de l'en-tête encapsulant. Sinon, la chaîne d'en-tête NE DOIT PAS être compressée.

Note : Si c'est l'en-tête IP le plus proche d'un en-tête TCP, le bit 7 du champ Classe de trafic doit être passé en utilisant le fanion R de l'en-tête TCP compressé. voir le paragraphe 6 a).

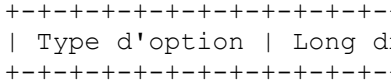
Cette classification implique que l'en-tête IPv6 de base entier sera compressé.

7.2 En-têtes d'extension IPv6

Les en-têtes d'extension sont décrits à la section 4 de la [RFC2460].
Les en-tête d'extension qui sont présents et leur ordre relatif ne sont pas supposés changer dans un flux de paquets. Chaque fois qu'il y a un changement, un en-tête complet de paquet doit être envoyé. Tous les champs Prochain en-tête dans les en-têtes IPv6 de base et dans les en-têtes d'extension IPv6 sont NOCHANGE.

7.3 Options

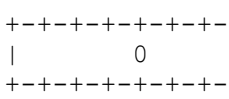
Le contenu des en-têtes d'extension Options bond par bond et Destination sont codés avec "options" LV (voir au paragraphe 4.2 de la [RFC2460]):



Les champs Type d'option et Longueur de données d'option sont supposés être fixes pour un flux de paquets donné, de sorte

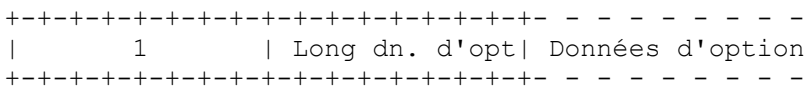
qu'ils sont classés NOCHANGE. Les données d'option sontALÉATOIRE sauf spécification contraire ci-dessous.

Bourrage
Option Pad1



L'option entière est NOCHANGE.

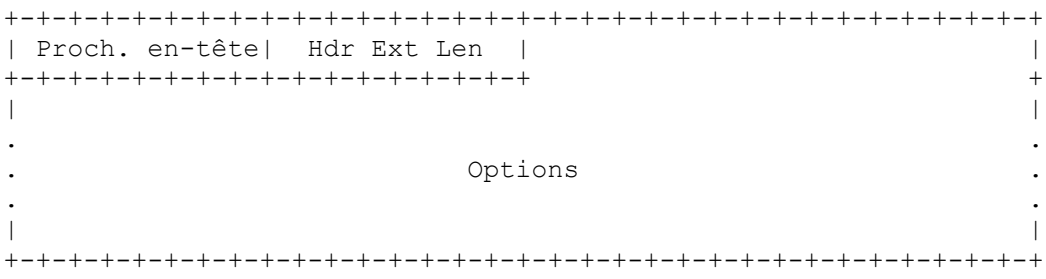
Option PadN



Tous les champs sont NOCHANGE.

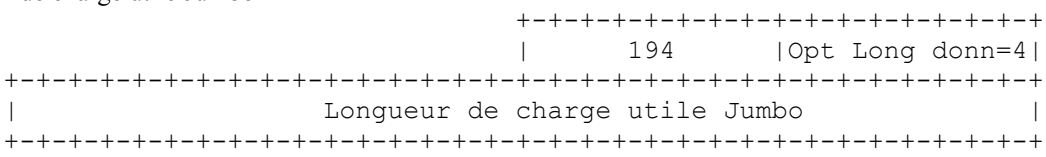
7.4 En-têtes d'options bond par bond

Les en-têtes d'options bond par bond sont décrits au paragraphe 4.3 de la [RFC2460].



Prochain en-tête NOCHANGE
Hdr Ext Len *(Longueur d'en-tête externe)* NOCHANGE
Options Valeurs codées en TLV et bourrage.
Classé conformément au paragraphe 7.3 ci-dessus, sauf cas d'une option de charge utile Jumbo (voir ci-dessous).

Option de charge utile Jumbo

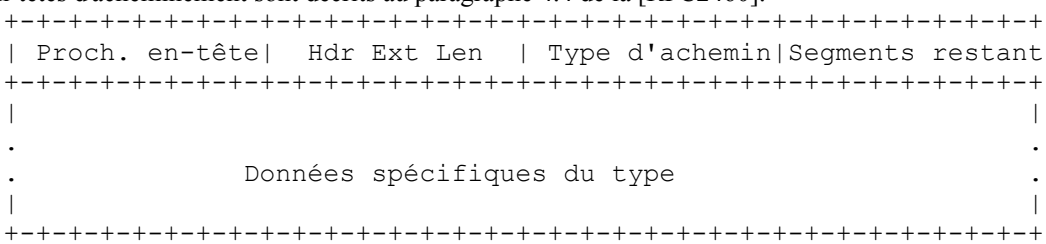


Les deux premiers champs sont NOCHANGE et Longueur de charge utile Jumbo est INFÉRÉ. (La longueur de trame doit être fournie par la mise en œuvre de couche liaison.)

Note : Il n'y a pas de sens à compresser les en-têtes d'un paquet qui porte une option de charge utile Jumbo car la redondance de l'en-tête qui s'y rapporte est négligeable. De plus, c'est habituellement une mauvaise idée d'envoyer des paquets d'une telle longueur sur des liaisons à faible ou moyenne vitesse.

7.5 En-tête d'acheminement

Les en-têtes d'acheminement sont décrits au paragraphe 4.4 de la [RFC2460].



Tous les champs de l'en-tête d'acheminement sont NOCHANGE.

Si le Type d'acheminement n'est pas reconnu, il est impossible de déterminer l'adresse de destination finale sauf si le champ Segments restant a la valeur zéro, auquel cas l'adresse de destination est l'adresse de destination finale dans l'en-tête IPv6 de base.

Dans l'en-tête d'acheminement de type 0, la dernière adresse est DEF si (Segments restant > 0).

Les en-têtes d'acheminement sont complètement compressés. C'est un gros gain car la taille maximum de l'en-tête d'acheminement est de 392 octets.De plus, les en-têtes d'acheminement de type 0 avec une adresse, de taille 24 octets, sont utilisées par IP mobile.

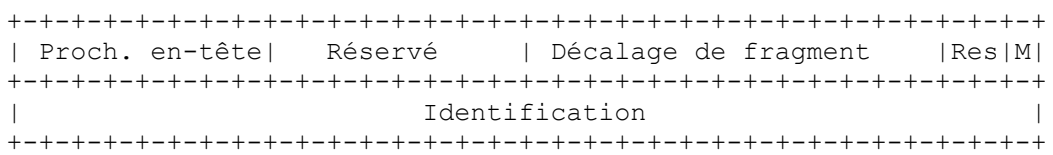
7.6 En-tête de fragment

Les en-têtes de fragment sont décrits au paragraphe 4.5 de la [RFC2460].
Le premier fragment d'un paquet a Décalage de fragment = 0 et la chaîne de sous en-têtes s'étend au-delà de son En-tête de fragment. Si un fragment n'est pas le premiert (Décalage de fragment différent de 0) il n'y a pas de sous en-têtes suivants (sauf si la chaîne de sous en-têtes dans le premier fragment ne tenait pas entièrement dans le premier fragment).

Comme les paquets peuvent être réordonnés avant d'atteindre le point de compression, et comme des fragments peuvent suivre d'autres routes à travers le réseau, un compresseur ne peut pas se fier au fait qu'il voit le premier fragment avant les autres fragments. Cela implique que les informations dans les sous en-têtes qui suivent l'En-tête de fragment du premier fragment ne peuvent pas être examinées pour déterminer le flux de paquets approprié pour les autres fragments.

Il est possible de concevoir des schémas de compression qui compressent les sous en-têtes après l'en-tête de fragment, au moins dans le premier fragment, mais pour éviter de compliquer les règles d'envoi des en-tête complets et les règles pour la compression et la décompression, la chaîne de sous en-têtes qui suit un en-tête de fragment NE DOIT PAS être compressée.

Les champs de l'en-tête de fragment sont classés comme suit.



Proch. en-tête	NOCHANGE
Réservé	NOCHANGE
Res	ALÉATOIRE
Fanion M	ALÉATOIRE
Décalage de fragment	ALÉATOIRE
Identification	ALÉATOIRE

Cette classification implique qu'un en-tête de fragment est compressé à 6 octets. La MTU minimum de IPv6 est 1280 octets de sorte que la plupart des fragments auront au moins 1280 octets. Comme la redondance de 6 octets de l'en-tête de fragment compressé est amortie sur un très gros paquet, la complexité supplémentaire de schémas de compression plus sophistiqués n'est pas justifiable.

Note : Le champ Identification est ALÉATOIRE au lieu de NOCHANGE pour éviter un démarrage lent de compression par paquet original.

Groupement des fragments conformément aux lignes directrices facultatives du paragraphe 4.1 :

Les fragments et les paquets non fragmentés ne devraient pas être regroupés.

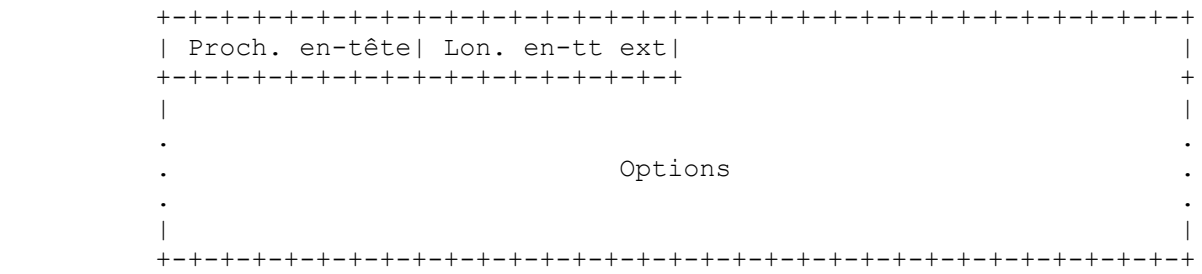
Les numéros d'accès ne peuvent pas être utilisés pour identifier le flux de paquets parce que les numéros d'accès ne sont pas présents dans tous les fragments. Pour coller aux règles d'unicité de la valeur d'identification, un flux de paquets fragmenté est identifié par la combinaison de l'adresse de source et de l'adresse de destination (finale).

Note : La valeur de Identification N'EST PAS utilisée pour identifier le flux de paquets. Cela évite d'utiliser un nouveau CID pour chaque paquet et économise le coût du démarrage lent de compression qui lui est associé. On prévoit que

la partie non fragmentable des en-têtes ne va pas changer trop fréquemment ; cela peut entraîner des ennuis si cela arrive.

7.7 En-tête d'options de destination

Les en-têtes d'options de destination sont décrits au paragraphe 4.5 de la [RFC2460].



Proch. en-tpete	NOCHANGE
Longueur d'en-tête externe	NOCHANGE
Options	Valeurs codées en TLV et bourrage. Compressé selon le § 7.3 ci-dessus.

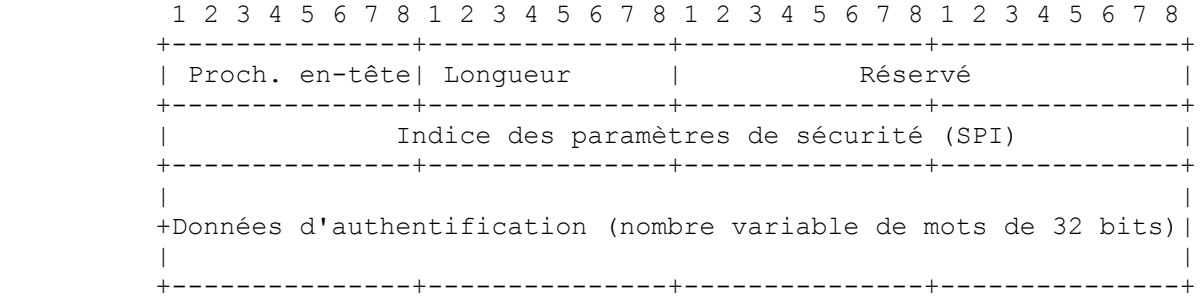
Les seules options Destination définies dans la [RFC2460] sont les options de bourrage.

7.8 Pas de prochain en-tête

Couvert par les règles pour les extensions d'en-tête IPv6 (paragraphe 7.2).

7.9 En-tête d'authentification

L'en-tête d'authentification est décrit au paragraphe 3.2 de la [RFC2402].



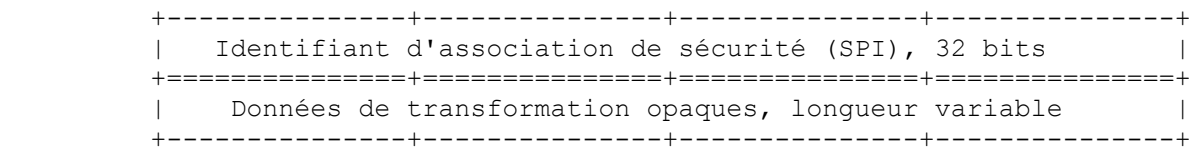
Proch. en-tête	NOCHANGE
Longueur	NOCHANGE
Réservé	NOCHANGE
SPI	NOCHANGE (DEF)
Données d'authentification	ALÉATOIRE

La [RFC1828] spécifie comment faire l'authentification avec les clés de MD5, méthode d'authentification que toute mise en œuvre de IPv6 doit prendre en charge. Pour cette méthode, Données d'authentification est de 16 octets.

7.10 Encapsulation de l'en-tête de charge utile de sécurité

L'encapsulation de l'en-tête de charge utile de sécurité est décrite au paragraphe 3.1 de la [RFC2406].
Cet en-tête implique que les parties suivantes du paquet sont chiffrées. Donc, aucune autre compression d'en-tête n'est possible sur les en-têtes suivants car le chiffrement est normalement déjà effectué lorsque le compresseur voit le paquet.

Cependant, lorsque l'en-tête ESP est utilisé en mode tunnel, un paquet IP entier est chiffré, et les en-têtes de ce paquet PEUVENT être compressés avant que le paquet ne soit chiffré au point d'entrée du tunnel. Cela signifie qu'il doit être possible de fournir un paquet IP et sa longueur au décompresseur, comme si il venait de la couche liaison. Les mécanismes pour traiter le réarrangement décrits à la section 11 DOIVENT aussi être utilisés, car les paquets peuvent être réarrangés dans un tunnel.



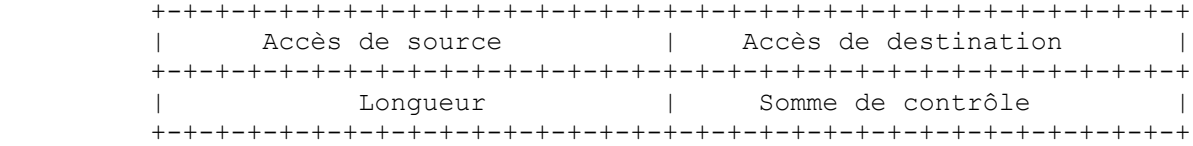
SPI	NOCHANGE (DEF)
Données de transformation opaques	ALÉATOIRE

Tout ce qui est après le SPI est chiffré et n'est pas compressé.

7.11 En-tête UDP

L'en-tête UDP est décrit dans la [RFC0768].

Le champ Prochain en-tête (IPv6) ou Protocole (IPv4) dans le sous en-tête précédent est DEF.



Accès de source	NOCHANGE (DEF)
Accès de destination	NOCHANGE (DEF)
Longueur	INFÉRÉ
Somme de contrôle	ALÉATOIRE, sauf si elle est zéro, auquel cas elle est NOCHANGE.

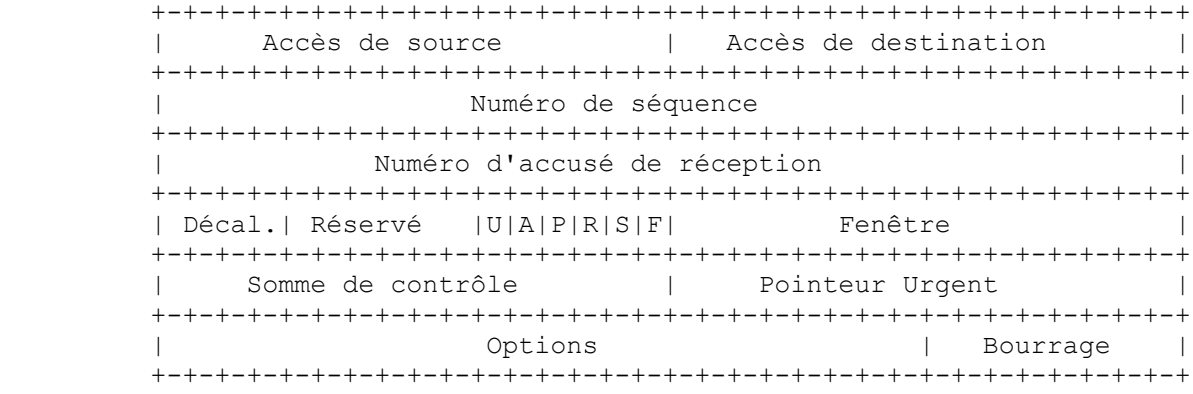
Le champ Longueur de l'en-tête UDP DOIT correspondre au ou aux champs Longueur des sous en-têtes précédents, c'est-à-dire qu'il ne doit pas y avoir de bourrage après la charge utile UDP qui est couverte par la Longueur IP.

L'en-tête UDP est normalement compressé à 2 octets, la somme de contrôle UDP. Lorsque la somme de contrôle UDP est zéro (ce qui ne peut être avec IPv6) cela sera vraisemblablement le cas pour tous les paquets du flux et est défini comme étant NOCHANGE. Cela économise 2 octets dans l'en-tête compressé.

7.12 En-tête TCP

L'en-tête TCP est décrit dans la [RFC0793].

Le champ Prochain en-tête (IPv6) ou Protocole (IPv4) dans le sous en-tête précédent est DEF.



U, A, P, R, S, et F signifient respectivement Urgent, Accusé de réception, Psh, Rst, Syn, et Fin.

Il y a deux façons de compresser l'en-tête TCP.

7.12.1 Compression avec codage différentiel

Accès de source	NOCHANGE (DEF)

Accès de destination		NOCHANGE (DEF)
Numéro de séquence		DELTA
N° d'accusé de réception		DELTA
Décalage		NOCHANGE
Réservé	DELTA	(si différent du contexte, établir le fanion R dans l'octet des fanions et envoyer la valeur absolue comme décrit en 6 a.)
Urg,Psh		ALÉATOIRE (placed in flag octet)
Ack		INFÉRÉ comme étant 1
Rst,Syn,Fin		INFÉRÉ comme étant 0
Fenêtre	DELTA	(si il y a un changement dans Fenêtre, mettre le fanion W dans l'octet Fanions et envoyer la différence)
Somme de contrôle		ALÉATOIRE
Pointeur Urgent		DELTA (si Urgent est mis, envoyer la valeur absolue)
Options, Bourrage		DELTA (si il y a un changement dans Options, mettre le fanion O et envoyer toutes les options, bourrage)

Un paquet avec un en-tête TCP compressé conformément à ce qui est mentionné ci-dessus doit être indiqué avec le type COMPRESSED_TCP. L'en-tête compressé est décrit à la section 6.

Cette méthode comporte essentiellement les techniques de codage différentiel de Jacobson, décrites dans la [RFC1144], les différences étant le placement des champs TCP de l'en-tête compressé (voir la section 6) l'utilisation du fanion O, du fanion R, et l'élimination du fanion C. Le fanion O permet la compression de l'en-tête TCP lorsque l'option Horodatage est utilisée et que les champs Options changent à chaque en-tête.

Les valeurs DELTA (sauf pour le champ Réservé, Options, et Bourrage) DOIVENT être codées comme dans la [RFC1144]. Une valeur de champ Réservé passée avec le fanion R NE DOIT PAS mettre à jour le contexte au compresseur ou au décompresseur.

7.12.2 Sans codage différentiel

Accès de source	NOCHANGE (DEF)
Accès de destination	NOCHANGE (DEF)
(tout le reste)	ALÉATOIRE

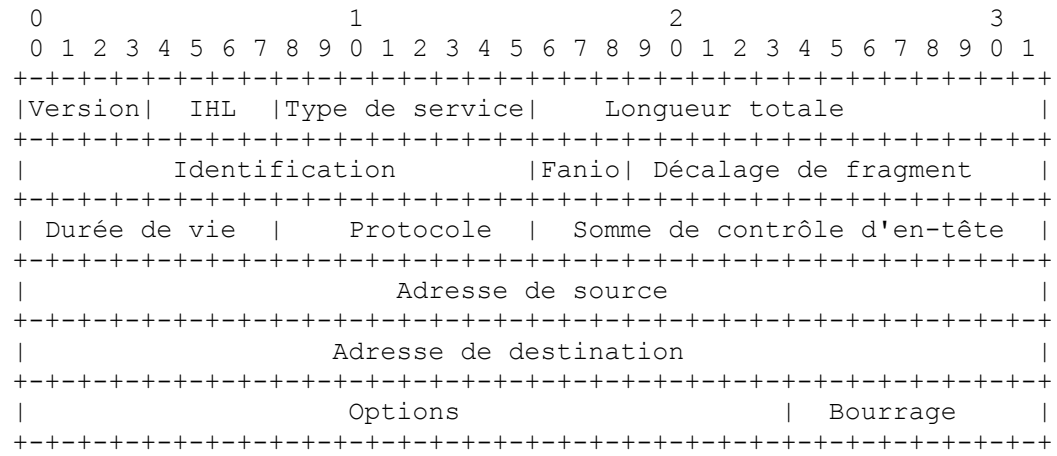
Le champ Identification dans un en-tête IPv4 précédent est ALÉATOIRE.

Un paquet avec un en-tête TCP compressé conformément à ce qui précède doit être indiqué avec le type COMPRESSED_TCP_NODELTA. Il utilise le même espace de CID que les paquets COMPRESSED_TCP, et l'en-tête DOIT être sauvegardé comme contexte. L'en-tête compressé est décrit à la section 6.

Ce type de paquet peut être envoyé en réponse à une demande d'en-tête au lieu d'envoyer un en-tête complet, et il peut être utilisé sur des liaisons qui réarrangent les paquets, et ils peuvent être envoyés à la place d'un en-tête complet lorsque il y a des changements qui ne peuvent pas être représentés par un en-tête compressé. Un compresseur sophistiqué peut se mettre à n'envoyer que des en-têtes COMPRESSED_TCP_NODELTA lorsque la fréquence de perte de paquet est élevée.

7.13 En-tête IPv4

L'en-tête IPv4 est décrit au paragraphe 3.1 de la [RFC0791].



Il y a deux façon de compresser l'en-tête IPv4 :

a) Si l'en-tête IPv4 n'est pas pour un fragment (le fanion MF n'est pas mis et Décalage de fragment est à zéro) et si il n'y a pas d'options (IHL est 5) il est classé comme suit :

Version	NOCHANGE	(DEF)
IHL	NOCHANGE	(DEF, must be 5)
Type de service	NOCHANGE	(Peut être DEF, voir au paragraphe 4.1) (voir aussi 6 a)
Longueur totale	INFÉRÉ	(D'après la mise en œjuvre de couche liaison ou l'en-tête IP encapsulant)
Identification	DELTA	(Si le champ Protocole a la valeur correspondant à TCP)
	ALÉATOIRE	(autrement)
Fanions	NOCHANGE	(Le fanion MF ne doit pas être mis)
Décalage de fragment	NOCHANGE	(Doit être zéro)
Durée" de vie	NOCHANGE	(Peut être DEF, voir au paragraphe 4.1)
Protocole	NOCHANGE	
Somme de contrôle d'en-tête	INFÉRÉ	(Calculée d'après les autres champs)
Adresse de source	NOCHANGE	(DEF)
Adresse de destination	NOCHANGE	(DEF)
Options, Bourrage	(non présent)	

Note : Lorsque un en-tête TCP suit immédiatement, les en-têtes IPv4 et TCP DOIVENT être compressés comme une unité, comme décrit à la section 6. Les bits 6 et 7 du champ Type de service (bits 14 et 15 du premier mot) peuvent alors être passés en utilisant le fanion R (voir la section 6 a).

b) Si l'en-tête IPv4 est pour un fragment (bit MF établi, ou Décalage de fragment différent de zéro) ou si il y a des options (IHL > 5) tous les champs sont à ALÉATOIRE (c'est-à-dire, si l'en-tête est compressé, tous les champs sont envoyés tels quels et non compressés). Cette classification permet la compression de l'en-tête de tunnel, mais pas de l'en-tête de fragment, lorsque les fragments sont tunnelés. Si l'en-tête IPv4 est pour un fragment, il termine la chaîne compressible des sous en-têtes, c'est-à-dire qu'il doit être le dernier sous en-tête à être compressé. Si l'en-tête IPv4 a des options mais n'est pas pour un fragment, il ne termine pas la chaîne des sous en-têtes compressibles, et donc les sous en-têtes suivants peuvent être compressés.

Un compresseur qui respecte les lignes directrices facultatives du paragraphe 4.1 va dans le cas a) utiliser Version, Adresse de source et Adresse de destination pour définir le flux de paquets, ainsi que le fait qu'il n'y a pas d'option IPv4 et que ce n'est pas un fragment.

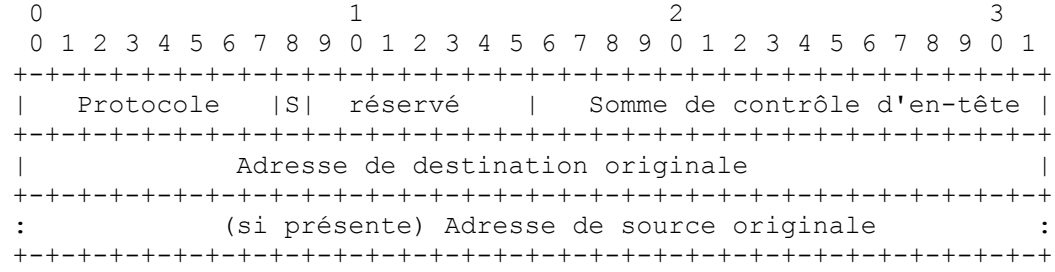
Le cas b) peut définir deux sortes de flux de paquets selon que l'en-tête IPv4 est ou non pour un fragment.

Si l'en-tête IPv4 dans le cas b) est pour un fragment, un compresseur qui suit les lignes directrices facultatives va utiliser ce fait ainsi que Version, Adresse de source, et Adresse de destination pour déterminer le flux de paquets.

Si l'en-tête IPv4 dans le cas b) n'est pas pour un fragment, il doit avoir des options. Un compresseur qui suit les lignes directrices facultatives va utiliser ce fait, mais pas la taille des options, ainsi que Version, Adresse de source, et Adresse de destination pour déterminer le flux de paquets.

7.14 En-tête d'encapsulation minimale

L'en-tête d'encapsulation minimale est décrit au paragraphe 3.1 de la [RFC2004].



Protocole	NOCHANGE
Adresse de source originale présente (S)	NOCHANGE
réservé	NOCHANGE
Somme de contrôle d'en-tête	INFÉRÉE (calculée d'après d'autres valeurs)
Adresse de destination originale	NOCHANGE
Adresse de source originale	NOCHANGE (présente seulement si S=1)

Cet en-tête sera vraisemblablement utilisé par IP mobile.

8. Changement des identifiants de contexte

Sur une liaison point à point, le compresseur a une connaissance complète des CID qui sont utilisés au décompresseur et peut changer le CID qu'utilise un flux de paquets ou réutiliser des CID à volonté.

Chaque CID non TCP est associé à un contexte par une valeur de génération. Pour éviter un retour à zéro trop rapide des générations et une éventuelle décompression incorrecte, une mise en œuvre DOIT éviter le retour à zéro de la valeur de génération en moins de MIN_WRAP secondes (voir la section 14).

Pour aider à éviter le retour à zéro, la valeur de génération associée à un CID NE DOIT PAS être remise à zéro lorsque on passe à un nouveau flux de paquets. Un compresseur DOIT plutôt incrémenter la valeur de génération de un lorsqu'il utilise le CID pour un nouveau flux de paquets non TCP.

9. Règles d'abandon ou de mémorisation temporaire des paquets

Lorsque un décompresseur reçoit un paquet avec un en-tête TCP compressé avec le CID C, il DOIT l'éliminer lorsque le contexte pour C n'a pas été initialisé par un en-tête complet.

Lorsque un décompresseur reçoit un paquet avec un en-tête non TCP compressé avec un CID C et une génération G, l'en-tête ne doit pas être décompressé en utilisant le contexte actuel lorsque :

a) le décompresseur a été déconnecté du compresseur depuis plus de MIN_WRAP secondes, parce que le context peut être obsolète même si il a la génération G ;
b) le contexte pour C a une génération autre que G.

Dans les cas a) et b) le paquet peut être :

i) éliminé immédiatement,
ii) mémorisé temporairement jusqu'à ce que le contexte soit mis à jour par un paquet avec un en-tête non TCP complet avec le CID C et la génération G, après quoi l'en-tête peut être décompressé.

Les paquets mémorisés de cette manière DOIVENT être éliminés lorsque :

*) on reçoit des en-tête non TCP complets ou compressés avec le CID C et la génération autre que G,
*) le décompresseur n'a pas reçu de paquets avec le CID C dans les dernières MIN_WRAP secondes.

Lorsque des en-têtes complets sont perdus, un décompresseur peut recevoir des en-têtes non TCP compressés avec une valeur de génération autre que la génération de son contexte. La règle ii) permet au décompresseur de mémoriser de tels en-têtes jusqu'à ce qu'ils puissent être décompressés en utilisant le contexte correst.

10. Compression d'en-tête à faible perte pou TCP

Comme moins de bits sont transmis par paquet avec la compression d'en-tête, le taux de perte de paquet est moins élevé avec la compression d'en-tête que sans, pour un taux d'erreurs binaire fixé. Ceci est bénéfique pour les liaisons avec des taux d'erreurs binaires élevé comme les liaisons sans fil.

Cependant, comme les en-têtes TCP sont compressés en utilisant un codage différenciel, un seul segment TCP perdu peut amener la ruine d'une fenêtre d'envoi TCP entière parce que le contexte n'est pas correctement incrémenté au décompresseur. Les en-têtes suivants vont donc être décompressés de façon différente de ce qu'ils étaient avant compression et éliminés par le receveur TCP à cause de l'échec de la somme de contrôle TCP.

Une connexion TCP dans une grande zone où le dernier bond est sur une liaison à perte à moyen débit, par exemple, un LAN sans fil, aura alors de faibles performances avec une compression d'en-tête traditionnelle parce que le délai produit par la bande passante est relativement élevé et le taux d'erreurs bianires relativement élevé. Pour un LAN sans fil à 2 Mbit/s et un RTT de bout en bout de 200 ms, le délai produit par la bande passante est de 50 koctets. C'est équivalent à environ 97 segments de 512 octets avec des en-têtes compressés. Chaque perte peut donc être multipliée par un facteur de 100.

La présente section décrit deux mécanismes simples de réparation rapide du contexte. Avec ces mécanismes, la compression d'en-tête va améliorer le débit de TCP sur les liaisons à perte aussi bien que sur les liaisons avec de faibles taux d'erreurs binaires.

10.1 Algorithme "deux fois"

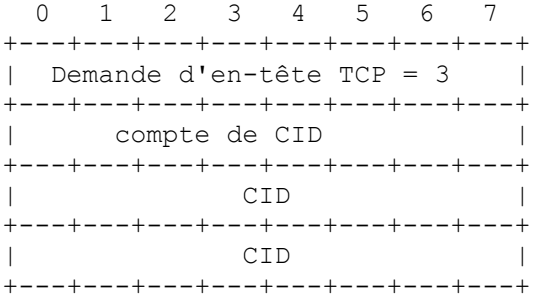
Le décompresseur peut calculer la somme de contrôle TCP pour déterminer si son contexte est mis à jour correctement. Si la somme de contrôle échoue, on suppose que l'erreur est causée par un segment perdu qui n'a pas mis correctement à jour le contexte. Le delta du segment actuel est alors ajouté au contexte, toujours avec l'hypothèse que le segment perdu contenait le même delta que l'actuel. En décompressant et en calculant à nouveau la somme de contrôle TCP, le décompresseur vérifie si la réparation a réussi ou si le delta devrait être appliqué une fois de plus.

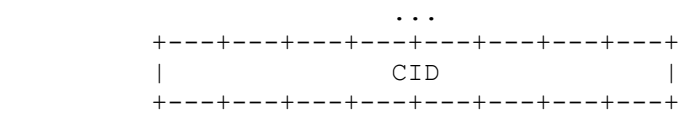
L'analyse des traces des divers transferts TCP en vrac montre que l'application une ou deux fois du delta au segment en cours va réparer le contexte pour entre 83 et 99 pour cent de toutes les pertes d'un seul segment dans le flux des données. Pour le flux d'accusés de réception, le taux de succès est inférieur du fait du mécanisme de retard d'accusé de réception de TCP. Le mécanisme "deux fois" répare le contexte dans 53 à 99 pour cent des pertes dans le flux d'accusés de réception. Une mise en œuvre sophistiquée de cette idée déterminerait si le flux TCP est d'accusé de réception ou de données et déterminerait la taille de segment en observant le flux d'en-têtes complets et compressés. Essayer les deltas qui sont de petits multiples de la taille de segment résulterait en des taux encore supérieurs de réussite de réparations pour les flux d'accusés de réception.

10.2 Demandes d'en-tête

Le taux de succès relativement faible de l'algorithme "deux fois" pour les flux d'accusés de réception TCP appelle un mécanisme supplémentaire pour réparer le contexte chez le décompresseur. Lorsque le décompresseur échoue à réparer le contexte après une perte, le décompresseur peut facultativement demander un en-tête complet au compresseur. C'est possible sur les liaisons où le décompresseur peut identifier le compresseur et lui envoyer des paquets.

Sur de telles liaisons, un décompresseur peut renvoyer un paquet CONTEXT_STATE au compresseur pour indiquer qu'un ou plusieurs contextes sont invalides. Un décompresseur NE DEVRAIT PAS transmettre un paquet CONTEXT_STATE chaque fois qu'un paquet compressé se réfère à un contexte invalide, mais devrait plutôt limiter le taux de transmission des paquets CONTEXT_STATE pour éviter d'inonder le canal inverse. Un paquet CONTEXT_STATE peut indiquer que plusieurs contextes sont périmés, cette technique DEVRAIT être utilisée plutôt que d'envoyer plusieurs paquets séparés. Le diagramme ci-après montre le format d'un paquet CONTEXT_STATE.





Le premier octet est un code de type pour permettre que le type de paquet CONTEXT_STATE soit partagé par les autres protocoles de compression qui sont (voir la [RFC2460]) ou peuvent être définis en parallèle avec celui-ci. Lorsque il est utilisé pour les demandes d'en-tête TCP, le code de type a la valeur 3, et le reste du paquet est une séquence de CID précédée par un compte de un octet du nombre de CID.

À réception d'un paquet CONTEXT_STATE, le compresseur DOIT marquer les CID comme invalides pour s'assurer que le prochain paquet émis dans ce flux de paquets est un paquet FULL_HEADER ou COMPRESSED_TCP_NODELTA.

Les demandes d'en-tête sont une optimisation, de sorte que la perte d'un paquet CONTEXT_STATE n'affecte pas le fonctionnement correct de la compression d'en-tête TCP. Lorsque un paquet CONTEXT_STATE est perdu, un nouveau peut éventuellement être transmis, ou TCP arrivera en fin de temporisation et retransmettra. Le gros avantage de l'utilisation des demandes d'en-têtes est que les flux d'accusés de réception TCP peuvent être réparés après un délai d'aller-retour sur la liaison à pertes. Cela va normalement éviter d'arriver à la fin de temporisation de TCP et à des retransmissions inutiles. Le plus faible taux de perte de paquet dû à de plus petits paquets va alors résulter en un plus fort débit parce que la fenêtre TCP peut augmenter entre les pertes.

11. Liaisons qui réarrangent les paquets

Certaines liaisons réarrangent les paquets, par exemple les liaisons radio multi-bonds qui utilisent l'acheminement par déflexion pour acheminer autour de nœuds encombrés. Des paquets acheminés par des voies différentes peuvent alors arriver à destination dans un ordre différent de celui de leur envoi.

11.1 Réarrangement dans les flux de paquets non TCP

Les en-têtes non TCP compressés ne changent pas de contexte, pas plus que les en-têtes complets qui le rafraîchissent.Il ne peut y avoir de problèmes que lorsque un en-tête complet qui change le contexte arrive décalé. Il y a deux cas :

- Un paquet avec un en-tête complet avec la génération G arrive *après* un paquet avec en-tête compressé de génération G. Ce cas est traité par la règle b) ii) à la section 9.
- Un paquet avec en-tête complet de génération G arrive *avant* un paquet avec en-tête compressé de génération G-1 (modulo 64). Le décompresseur PEUT alors garder les deux versions du contexte pendant quelques temps pour être capable de décompresser les en-têtes compressés suivants de génération G-1 (modulo 64). Le vieux contexte DOIT être éliùminé après MIN_WRAP secondes.

11.2 Réarrangement dans les flux de paquets TCP

Un compresseur peut éviter d'envoyer des en-têtes COMPRESSED_TCP et envoyer seulement des en-têtes COMPRESSED_TCP_NODELTA lorsque il y a des réarrangements sur la liaison. Les en-têtes compressés vont normalement être de 17 octets avec cette méthode, significativement plus grands que les 4 à 7 octets usuels.

Pour réaliser de meilleurs taux de compression, la méthode suivante, qui ajoute seulement deux octets à l'en-tête compressé pour un total de 6 à 9 octets, peut être utilisée. Un numéro de séquence de paquet, incrémenté de un à chaque paquet dans le flux TCP, est alors associé à chaque en-tête compressé et complet. Cela permet au décompresseur de placer les paquets dans la séquence correcte et d'appliquer leurs deltas au contexte dans l'ordre correct. Un simple schéma de fenêtre glissante est utilisé pour placer les paquets dans l'ordre correct.

Deux octets sont nécessaires pour les numéros de séquence du paquet. Un octet ne donne que 256 numéros de séquence. Dans un schéma de fenêtre glissante, la fenêtre ne devrait pas faire plus de la moitié de l'espace de numéros de séquence, de sorte que les paquets ne peuvent pas arriver de plus de 127 positions hors séquence. Ceci est équivalent à un délai de 260 ms sur des liaisons à 2 Mbit/s avec des segments de 512 octets. Les délais de cet ordre ne sont pas rares sur des connexions Internet de large zone. Cependant, deux octets donnant 2^16 = 65 536 valeurs devrait suffire.

Les en-têtes TCP/IP complets n'auront d'espace que pour un octet de numéro de séquence lorsque il n'y a pas de tunnelage. Il n'est pas faisable d'augmenter la taille des en-têtes complets car la taille de paquet peut être optimisée pour la MTU de la liaison. Donc, seul l'octet de moindre poids du numéro de séquence du paquet peut être placé dans de tels en-têtes complets. On pense que de tels en-têtes complets peuvent être positionnés correctement suffisamment fréquemment avec seulement l'octet de moindre poids du numéro de séquence du paquet disponible.

Le numéro de séquence de paquet zéro DOIT être sauté. Éviter le zéro tient compte d'un problème qui peut survenir lorsque l'option d'adaptation de la fenêtre TCP est utilisée pour élargir la fenêtre TCP. Lorsque exactement 2^16 octets de données TCP sont perdues, un en-tête compressé sera décompressé incorrectement sans être détecté par la somme de contrôle TCP. Les tailles de segment TCP sont souvent une puissance de deux. En utilisant ainsi un espace de numéro de séquence de paquet qui ne soit pas une puissance de deux, le numéro de séquence TCP ou le numéro de séquence du paquet vont différer lorsque 2^16 octets sont perdus. Chaque fois qu'un compresseur voit l'option d'adaptation de fenêtre sur un segment SYN, il DOIT utiliser les numéros de séquence de paquet lorsque il compresse ensuite ce flux de paquets.

Dans les en-têtes TCP compressés, le numéro de séquence de paquet de deux octets DOIT être placé immédiatement après la somme de contrôle TCP. Voir au paragraphe 5.3 le placement des numéros de séquence de paquets dans les en-têtes complets.

12. Clés pour une compression d'en-tête supplémentaire

On fournit l'astuce suivante pour permettre des schémas supplémentaires de compression d'en-tête pour les en-têtes par dessus UDP. La chaîne initiale de sous en-têtes est alors compressée comme décrit plus haut, et l'autre schéma de compression d'en-tête est appliqué à l'en-tête par dessus l'en-tête UDP. Un exemple d'une telle compression d'en-tête supplémentaire est le RTP Compressé par Casner et Jacobson [RFC2460]. Pour permettre une certaine détection d'erreur, de tels schémas ont normalement besoin d'un numéro de séquence qui peut devoir être passé dans des en-têtes complets aussi bien que dans des en-têtes UDP compressés.

Le bit D et l'octet Données (voir la section 6) fournissent le mécanisme nécessaire. Lorsque un numéro de séquence, doit être passé dans un en-tête FULL_HEADER ou COMPRESSED_NON_TCP, le bit D est mis et le numéro de séquence est placé dans le champ Données. Le décompresseur doit alors extraire le champ Données et le rendre disponible pour le schéma supplémentaire de compression d'en-tête.

L'utilisation de schémas supplémentaires de compression d'en-tête comme CRTP doit être négociée. Le bit D et le mécanisme de l'octet Données doivent être activés automatiquement chaque fois que l'utilisation de schémas supplémentaires de compression d'en-tête a été négociée.

13. Démultiplexage

Pour chaque couche liaison, il doit y avoir un document qui spécifie comment sont indiqués les divers types de paquet utilisés par la compression d'en-tête IP. Un tel document existe pour PPP [RFC2460]. La présente section donne des lignes directrices FACULTATIVES sur la façon dont les types de paquet peuvent être indiqués par une couche de liaison particulière.

Il est nécessaire de distinguer les paquets qui ont des en-têtes IPv4 réguliers, des en-têtes IPv6 réguliers, les paquets IPv6 complets, les paquets IPv4 complets, les paquets TCP compressés, les paquets non TCP compressés, et les paquets CONTEXT_STATE.

La décision d'utiliser un ethertype (ou équivalent) distinct pour IPv6 a déjà été prise, ce qui signifie que les couches liaison doivent être capables d'indiquer qu'un paquet est un paquet IPv6.

La compression d'en-tête IP exige que la mise en œuvre de couche liaison puisse indiquer quatre sortes de paquets : COMPRESSED_TCP pour le format a) de la section 6, COMPRESSED_TCP_NODELTA pour le format b), COMPRESSED_NON_TCP pour les formats c) et d), et CONTEXT_STATE comme décrit au paragraphe 11.2. Il est aussi souhaitable d'indiquer FULL_HEADERS à la couche liaison.

Les en-têtes complets peuvent être indiqués en établissant le premier bit du champ Version dans un paquet indiqué comme étant un paquet IPv6. De plus, un bit du champ Version est utilisé pour indiquer si le premier sous en-tête est IPv6 ou IPv4, et un bit est utilisé pour indiquer si cet en-tête complet porte un CID TCP ou non TCP. Les quatre premiers bits sont codés comme suit :

Version	**Signification**
0110	en-tête IPv6 régulier
1T*0	T=1 indique un en-tête TCP, T=0 indique un en-tête non TCP
1*V0	V=1 indique un en-tête IPv6, V=0 indique un en-tête IPv4

Si une couche liaison ne peut pas indiquer les types de paquet pour les en-têtes compressés ou CONTEXT_STATE, les types de paquet qui ne peuvent pas être indiqués pourraient débuter par un octet indiquant le type de paquet, suivi par l'en-tête.

Premier octet	Type d'en-tête compressé
0	COMPRESSED_TCP
1	COMPRESSED_TCP_NODELTA
2	COMPRESSED_NON_TCP
3	CONTEXT_STATE

Les valeurs de type CONTEXT_STATE actuellement allouées sont :

Valeur	Type	Référence
0	Réservé	-
1	IP/UDP/RTP w. 8-bit CID	[RFC2460]
2	IP/UDP/RTP w. 16-bit CID	[RFC2460]
3	Demande d'en-tête TCP	paragraphe 10.2

14. Paramètres de configuration

Les paramètres de compression d'en-tête sont négociés d'une façon spécifique de la mise en œuvre de la couche de liaison. De telles procédures pour la couche liaison xxx seront spécifiées dans un document "Compression d'en-tête IP sur xxx". Un tel document existe pour PPP [RFC2460].

Les paramètres suivants sont fixés pour toutes les mises en œuvre du présent schéma de compression d'en-tête.

MIN_WRAP – Durée minimum du retour à zéro de la valeur de génération : 3 secondes.

Les paramètres suivants peuvent être négociés entre le compresseur et le décompresseur. Si il n'y a pas de négociation, leurs valeurs doivent être comme spécifié par défaut.

F_MAX_PERIOD – Plus grand nombre d'en-têtes non TCP compressés qui peut être envoyé sans envoi d'un en-tête complet.
Par défaut : 256
F_MAX_PERIOD doit être au moins 1 et au plus 65 535.

F_MAX_TIME – Les en-têtes compressés ne doivent pas être envoyés plus de F_MAX_TIME secondes après l'envoi du dernier en-tête complet.
Par défaut : 5
F_MAX_TIME doit être au moins de 1 et au plus de 255.

Note : F_MAX_PERIOD et F_MAX_TIME devraient être au plus bas quand il est probable qu'un décompresseur perd son état.

MAX_HEADER – Plus grande taille d'en-tête en octets qui peut être compressée.
Par défaut : 168 octets, ce qui couvre :
- deux en-têtes IPv6 de base,
- un en-tête d'authentification MD5 chiffré,
- un en-tête TCP de taille maximale.

MAX_HEADER doit être au moins de 60 octets et au plus de 65 535 octets.

TCP_SPACE – Valeur maximum du CID pour TCP.
Par défaut : 15 (ce qui donne 16 valeurs de CID)
TCP_SPACE doit être au moins de 3 et au plus de 255.

NON_TCP_SPACE – Valeur maximum du CID pour non TCP.
Par défaut : 15 (ce qui donne 16 valeurs de CID)
NON_TCP_SPACE doit être au moins de 3 et au plus de 65 535.

EXPECT_REORDERING – Les mécanismes de la section 11 sont utilisés.
Pas de valeur par défaut.

15. État de mise en œuvre

Un prototype qui utilise UDP comme couche de liaison fonctionne depuis mars 1996. Une mise en œuvre NetBSD pour PPP fonctionne depuis octobre 1996.

16. Remerciements

Le présent protocole utilise de nombreuses idées qui ont leur origine dans le concept de compression d'en-tête pour TCP/IP sur liaisons à basse vitesse de la [RFC1144] de Van Jacobson . Il a bénéficié de discussions avec Stephen Casner et Carsten Bormann.

Nous remercions Craig Partridge d'avoir mis le doigt sur un problème qui peut survenir lorsque l'option d'adaptation de fenêtre TCP est utilisée. Une solution à ce problème qui s'appuie sur les numéros de séquence des paquets utilisés pour le réarrangement est décrite au paragraphe 11.2.

17. Considérations pour la sécurité

Les protocoles de compression dans le présent document fonctionnent par dessus un protocole de couche liaison. Les protocoles de compression eux-mêmes n'introduisent pas de nouvelles faiblesses au delà de celles associées aux technologies spécifiques de la couche de liaison utilisées.

Les attaques de déni de service sont possibles si un intrus peut introduire (par exemple) des paquets à en-tête complet comportant des erreurs sur la liaison. Cependant, un intrus qui a la capacité d'injecter des paquets arbitraires à la couche de liaison de cette manière soulève de tels problèmes de sécurité que ceux en rapport avec l'utilisation de la compression d'en-tête paraissent minuscules en comparaison.

Nous conseillons que les mises en œuvre se prémunissent contre l'identification des flux de paquets à l'aide du chiffrement des informations, même si de telles informations sont disponibles au compresseur. Le faire peut exposer les schémas de trafic.

18. Adresse des auteurs

Mikael Degermark
Computer Science and Electrical Engineering
Lulea University of Technology
SE-971 87 Lulea, Sweden
téléphone : +46 920 91188
fax : +46 920 72831
mobile : +46 70 833 8933
mél : micke@sm.luth.se

Bjorn Nordgren
CDT/Telia Research AB
Aurorum 6
S-977 75 Lulea, Sweden
téléphone : +46 920 75400
fax : +46 920 75490
mél : bcn@lulea.trab.se,
bcn@cdt.luth.se

Stephen Pink
Computer Science and Electrical Engineering
Lulea University of Technology
SE-971 87 Lulea, Sweden
téléphone : +46 920 752 29
fax : +46 920 728 31
mobile : +46 70 532 0007
mél : steve@sm.luth.se

19. Références

[RFC0768] J. Postel, "Protocole de datagramme d'utilisateur", (STD 6), 28 août 1980.

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du protocole du programme Internet", STD 5, septembre 1981.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", (STD 7), septembre 1981.

[RFC1144] V. Jacobson, "Compression des en-têtes TCP/IP pour les liaisons série à faible débit", février 1990.

[RFC1553] S. Mathur et M. Lewis, "Compression d'en-têtes IPX sur support WAN (CIPX)", décembre 1993. *(Historique)*

[RFC1700] J. Reynolds et J. Postel, "Numéros alloués", STD 2, octobre 1994. *(Historique)* Voir à http://www.iana.org/numbers.html

[RFC1828] P. Metzger et W. Simpson, "Authentification IP avec du MD5 à clés", août 1995.

[RFC2004] C. Perkins, "Encapsulation minimale au sein de IP", octobre 1996. *(P.S.)*

[RFC2402] S. Kent et R. Atkinson, "En-tête d'authentification IP", novembre 1998. (*Obsolète, voir* RFC4302, 4305)

[RFC2406] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", novembre 1998. *(Obsolète, voir RFC4303)*

[RFC2460] S. Deering et R. Hinden, "Spécification du protocole Internet, version 6 (IPv6) ", décembre 1998. *(MàJ par RFC5095, D.S)*

[RFC2463] A. Conta, S. Deering, "Protocole de message de contrôle Internet (ICMPv6) pour le protocole Internet version 6 (IPv6)", décembre 1998. (*Obsolète, voir* RFC4443) *(D.S.)*

[RFC2508] S. Casner, V. Jacobson, "Compression d'en-têtes IP/UDP/RTP pour liaisons séries à bas débit", février 1999. *(P.S.)*

[RFC2509] M. Engan, S. Casner, C. Bormann, "Compression d'en-tête IP sur PPP", février 1999. (*Obsolète, voir* RFC3544) *(P.S.)*

20. Déclaration de droits de reproduction

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.