

Groupe de travail Réseau  
**Request for Comments : 2508**  
 Catégorie : En cours de normalisation  
 Traduction Claude Brière de L'Isle

S. Casner, Cisco Systems  
 V. Jacobson, Cisco Systems  
 février 1999

## Compression des en-têtes IP/UDP/RTP pour les liaisons de série à bas débit

### Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de copyright

Copyright (C) The Internet Society (1999). Tous droits réservés

### Résumé

Le présent document décrit une méthode pour compresser les en-têtes des datagrammes IP/UDP/RTP pour réduire la redondance sur les liaisons de série à bas débit. Dans de nombreux cas, les trois en-têtes peuvent être compressés à 2-4 octets.

On invite à faire des commentaires qui devraient être adressés à la liste de diffusion du groupe de travail rem-conf@es.net et/ou aux auteurs.

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

### Table des Matières

<u>1. Introduction.....</u>	<u>1</u>
<u>2. Hypothèses et compromis.....</u>	<u>2</u>
2.1 Unidirectionnel ou bidirectionnel.....	2
2.2 Segmentation et mise en couches.....	2
3. Algorithme de compression.....	3
3.1 Idée de base.....	3
3.2 Compression d'en-tête pour paquets de données RTP.....	3
3.3 Protocole.....	4
3.4 Compression des paquets de contrôle RTCP.....	11
3.5 Compression de paquets UDP non RTP.....	12
4. Interaction avec la segmentation.....	12
5. Négociation de la compression.....	12
6. Remerciements.....	12
7. Références.....	12
8. Considérations pour la sécurité.....	13
9. Adresse des auteurs.....	13
10. Déclaration complète de droits de reproduction.....	13

## 1. Introduction

Depuis la publication du protocole de transport en temps réel comme RFC [RFC1889], il y a eu un intérêt croissant pour l'utilisation de RTP comme étape pour réaliser l'interopérabilité entre des mises en œuvres différentes d'applications audio/vidéo sur le réseau. Cependant, il y a un souci avec l'en-tête RTPn de 12 octets qui donne trop de redondance pour les charges utiles de vingt octets lorsque on fonctionne sur des lignes à bas débit comme le font les modems à numérotation à 14,4 ou 28,8 kbit/s. (Certaines applications existantes qui fonctionnent dans cet environnement utilisent un protocole spécifique de l'application avec un en-tête de quelques octets qui a des fonctionnalités réduites par rapport à RTP.)

La taille de l'en-tête peut être réduite par des techniques de compression comme il a été fait avec grand succès pour TCP

[RFC1144]. Dans ce cas, la compression peut être appliquée à l'en-tête RTP seul, de bout en bout, ou à la combinaison des en-têtes IP, UDP et RTP sur la base de la liaison. Compresser les 40 octets d'en-têtes combinés procure un gain substantiellement plus important que de compresser les seuls 12 octets de l'en-tête RTP parce que la taille résultante est approximativement la même (2-4 octets) dans l'un ou l'autre cas. Compresser sur la base de la liaison donne aussi de meilleures performances parce que le délai et le taux de perte sont inférieurs. Donc, la méthode définie ici est pour la compression combinée des en-têtes IP, UDP et RTP sur la base de la liaison.

Le présent document définit un schéma de compression qui peut être utilisé avec IPv4, IPv6 ou des paquets encapsulés avec plus d'un en-tête IP, bien qu'il ait été initialement centré sur IPv4. La compression IP/UDP/RTP définie ici est destinée à tenir dans le cadre plus général de compression spécifié dans la [RFC2507] pour être utilisé avec IPv6 et IPv4. Ce cadre définit TCP et non TCP comme deux classes de transport au dessus de IP. La présente spécification crée IP/UDP/RTP comme troisième classe extraite de la classe non TCP.

## 2. Hypothèses et compromis

Le but de ce schéma de compression est de réduire les en-têtes IP/UDP/RTP à deux octets pour la plupart des paquets dans le cas où il n'y a pas d'envoi de somme de contrôle UDP ou à quatre octets avec les sommes de contrôle. Il est motivé principalement par le problème spécifique de l'envoi d'audio et de vidéo sur des modems à numérotation à 14,4 et 28,8. Ces liaisons tendent à fournir une communication bidirectionnelle, de sorte que le protocole tire parti de ce fait, bien que le protocole puisse aussi être utilisé avec des performances réduites sur des liaisons unidirectionnelles. Ce schéma de compression a de meilleures performances sur des liaisons locales avec un faible délai d'aller-retour.

La présente spécification ne traite pas de la segmentation et de la préemption des gros paquets pour réduire le délai sur les lignes à bas débit rencontré par les petits paquets en temps réel, sauf pour identifier à la Section 4 certaines interactions qui peuvent se produire entre segmentation et compression. Des schémas de segmentation peuvent être définis séparément et utilisés conjointement avec la compression définie ici.

On devrait noter que la simplicité de mise en œuvre est un facteur important à considérer dans l'évaluation d'un schéma de compression. Les serveurs de communications peuvent avoir besoin de prendre en charge la compression sur peut-être 100 lignes de modem à numérotation en utilisant un seul processeur. Il peut donc être approprié de faire quelques simplifications de la conception aux dépens de sa généralité, ou de produire un concept souple qui soit général mais puisse être subdivisé pour le simplifier. Un plus fort gain de compression pourrait être réalisé en faisant communiquer des modèles plus complexes pour les champs d'en-tête qui changent du compresseur au décompresseur, mais cette complexité est jugée non nécessaire. Les paragraphes qui suivent exposent certains des compromis cités ici.

### 2.1 Unidirectionnel ou bidirectionnel

En l'absence d'autres contraintes, un schéma de compression qui a fonctionné sur des liaisons unidirectionnelles serait préféré à un qui ne le faisait pas. Cependant, le fonctionnement sur une liaison unidirectionnelle exige des rafraîchissements périodiques avec un en-tête de paquet non compressé pour restaurer l'état de compression en cas d'erreur. Si un signal d'erreur explicite peut être retourné à la place, le délai de récupération peut être substantiellement raccourci. La redondance est aussi réduite dans le cas de non erreur. Pour obtenir ces améliorations de performances, la présente spécification inclut une indication d'erreur explicite envoyée sur le chemin inverse.

Sur une liaison unidirectionnelle, il serait possible d'utiliser à la place un rafraîchissement périodique. Chaque fois que le décompresseur détecte une erreur dans un flux de paquets particulier, il va simplement éliminer tous les paquets dans ce flux jusqu'à ce que soit reçu un en-tête non compressé pour ce flux, et reprendre ensuite la décompression. Le prix à payer serait le nombre potentiellement grand de paquets éliminés. La méthode du rafraîchissement périodique décrite au paragraphe 3.3 de la [RFC2507] s'applique à la compression de IP/UDP/RTP sur les liaisons unidirectionnelles ou les liaisons à fort délai ainsi qu'aux autres flux de paquets non TCP.

### 2.2 Segmentation et mise en couches

Le délai induit par le temps exigé pour envoyer un gros paquet sur la liaison lente n'est pas un problème pour l'audio unidirectionnel, par exemple, parce que le receveur peut s'adapter à la variance du délai. Cependant, pour les conversations interactives, minimiser le délai de bout en bout est critique. La segmentation de gros paquets, non en temps réel pour permettre la transmission de petits paquets en temps réel entre les segments peut réduire le délai.

La présente spécification ne traite que de la compression et suppose que la segmentation, si elle est incluse, sera traitée par une couche séparée. Il serait inapproprié d'intégrer la segmentation et la compression d'une façon telle que la compression

ne pourrait pas être utilisée par elle-même dans des situations où la segmentation était réputée inutile ou impraticable. De façon similaire, on aimerait éviter des exigences pour un protocole de réservation. Le schéma de compression peut être appliqué en local sur les deux extrémités d'une liaison indépendamment de tout autre mécanisme excepté l'exigence que la couche liaison fournisse des codes de type de paquet, une indication de longueur de paquet, et une bonne détection d'erreur.

À l'inverse, compresser séparément les couches IP/UDP et RTP perd trop des gains de compression qui sont possibles en les traitant ensemble. Franchir ces limites de couche de protocole est approprié parce que la même fonction est appliquée sur toutes les couches.

### 3. Algorithme de compression

L'algorithme de compression défini dans le présent document emprunte beaucoup à la conception de la compression d'en-tête TCP/IP décrite dans la [RFC1144]. Les lecteurs se référeront à cette RFC pour plus d'informations sur les motivations sous-jacentes et les principes généraux de la compression d'en-tête.

#### 3.1 Idée de base

Dans la compression d'en-tête TCP, la première réduction d'un facteur de deux en taux de données provient de l'observation que la moitié des octets dans les en-têtes IP et TCP restent constants sur la durée de vie de la connexion. Après l'envoi une fois de l'en-tête non compressé, ces champs peuvent être éliminés des en-têtes compressés qui suivent. La compression restante provient du codage différentiel sur les champs changeants pour réduire leur taille, et de l'élimination complète des champs changeants pour les cas courants en calculant les changements à partir de la longueur du paquet. Cette longueur est indiquée par le protocole de niveau liaison.

Pour la compression d'en-tête RTP, certaines des mêmes techniques peuvent être appliquées. Cependant, la plus grande partie du gain vient de l'observation que bien que plusieurs champs changent dans chaque paquet, la différence de paquet à paquet est souvent constante et que donc la différence de second ordre est zéro. En conservant à la fois l'en-tête non compressé et les différences de premier ordre dans l'état de session partagé entre le compresseur et le décompresseur, tout ce qui doit être communiqué est une indication que la différence de second ordre était zéro. Dans ce cas, le décompresseur peut reconstruire l'en-tête original sans aucune perte d'information simplement en ajoutant les différences de premier ordre à l'en-tête non compressé sauvegardé lorsque chaque paquet compressé est reçu.

Tout comme la compression d'en-tête TCP/IP entretient l'état partagé pour plusieurs connexions TCP simultanées, cette compression IP/UDP/RTP DEVRAIT conserver l'état pour plusieurs contextes de session. Un contexte de session est défini par la combinaison des adresses IP de source et de destination, des accès UDP de source et de destination, et du champ SSRC de RTP. Une mise en œuvre de compresseur peut utiliser une fonction de hachage sur ces champs pour indexer un tableau des contextes de session mémorisés. Le paquet compressé porte un petit entier, appelé identifiant de contexte de session ou CID, pour indiquer dans quel contexte de session un certain paquet devrait être interprété. Le décompresseur peut utiliser le CID pour indexer directement son tableau des contextes de session mémorisés.

Parce que la compression RTP est sans perte, elle peut être appliquée à tout trafic UDP qui en bénéficie. Très vraisemblablement, les seuls paquets qui vont en bénéficier sont des paquets RTP, mais il est acceptable d'utiliser des heuristiques pour déterminer si le paquet est ou non un paquet RTP parce qu'il n'y a pas de dommage à ce que l'heuristique donne une mauvaise réponse. Cela exige d'exécuter l'algorithme de compression pour tous les paquets UDP, ou au moins pour ceux qui ont des numéros d'accès pairs (voir au paragraphe 3.4).

La plupart des mises en œuvre de compresseur auront besoin de conserver une "mémoire tampon négative" des flux de paquets qui ont échoué à la compression comme paquets RTP pendant un certain nombre de tentatives afin d'éviter d'autres essais. Échouer à compresser signifie que certains champs de l'en-tête RTP potentiel dont il est attendu qu'ils restent constants la plupart du temps, comme le champ Type de charge utile, ont continué de changer. Même si les autres champs de ce genre restent constants, un flux de paquets qui a un champ SSRC qui change constamment DEVRAIT être entré dans la mémoire tampon négative pour éviter de consommer tous les contextes de session disponibles. La mémoire tampon négative est indexée par les paires d'adresse IP de source et de destination et d'accès UDP mais pas le champ SSRC RTP car ce dernier peut être changeant. Lorsque la compression RTP échoue, les en-têtes IP et UDP PEUVENT quand même être compressés.

Les paquets IP fragmentés qui ne sont pas des fragments initiaux et les paquets qui ne sont pas assez longs pour contenir un en-tête UDP complet NE DOIVENT PAS être envoyés comme paquets FULL\_HEADER. De plus, les paquets qui ne contiennent pas au moins 12 octets de données UDP NE DOIVENT PAS être utilisés pour établir un contexte RTP. Si un tel paquet est envoyé comme paquet FULL\_HEADER, il PEUT être suivi par des paquets COMPRESSED\_UDP mais NE DOIT PAS être suivi par des paquets COMPRESSED\_RTP.

### 3.2 Compression d'en-tête pour paquets de données RTP

Dans l'en-tête IPv4, seuls les champs Longueur totale, Identifiant de paquet, et Somme de contrôle d'en-tête vont normalement changer. La longueur totale est redondante avec la longueur fournie par la couche liaison, et comme ce schéma de compression doit dépendre de la couche de liaison pour fournir une bonne détection d'erreur (par exemple, le CRC de PPP [RFC1661]) la somme de contrôle d'en-tête peut aussi être éliminée. Cela ne laisse que l'identifiant de paquet, qui, en supposant qu'il n'y a pas de fragmentation IP, n'aura pas besoin d'être communiqué. Cependant, afin de conserver une compression sans perte, les changements de l'identifiant de paquet seront transmis. L'identifiant de paquet s'incrémente normalement de un ou d'un petit nombre pour chaque paquet. (Certains systèmes incrémentent l'identifiant avec les octets échangés, d'où il résulte un peu moins de compression.) Dans l'en-tête IPv6 de base il n'y a pas d'identifiant de paquet ni de somme de contrôle d'en-tête et seul le champ Longueur de charge utile change.

Dans l'en-tête UDP, le champ Longueur est redondant avec le champ Longueur totale IP et la longueur indiquée par la couche de liaison. Le champ Somme de contrôle UDP sera un zéro constant si la source choisit de ne pas générer de sommes de contrôle UDP. Autrement, la somme de contrôle doit être communiquée intacte afin de préserver la compression sans perte. Conserver une détection d'erreur de bout en bout pour les applications qui l'exigent est un principe important.

Dans l'en-tête RTP, l'identifiant de SSRC est constant dans un certain contexte car il fait partie de ce qui identifie le contexte particulier. Pour la plupart des paquets, seuls le numéro de séquence et l'horodatage vont changer d'un paquet à l'autre. Si les paquets ne sont pas perdus ou déclassés en amont du compresseur, le numéro de séquence va augmenter de un à chaque paquet. Pour les paquets audio de durée constante, l'horodatage va augmenter du nombre de périodes d'échantillonnage convoyées dans chaque paquet. Pour la vidéo, l'horodatage va changer sur le premier paquet de chaque trame, mais rester ensuite constant pour tout paquet supplémentaire de la trame. Si chaque trame vidéo occupe seulement un paquet, mais qu'elles sont générées à un taux constant, là encore, le changement d'horodatage de trame à trame est constant. Noter que dans chacun de ces cas, la différence de second ordre des champs de numéro de séquence et d'horodatage est zéro, de sorte que le prochain en-tête de paquet peut être construit d'après le précédent en-tête de paquet en ajoutant les différences de premier ordre pour les champs qui sont mémorisés dans le contexte de session avec le précédent en-tête non compressé. Lorsque la différence de second ordre n'est pas zéro, la magnitude du changement est habituellement plus petite que le nombre total de bits du champ, de sorte que la taille peut être réduite en codant la nouvelle différence de premier ordre et en la transmettant plutôt que la valeur absolue.

Le bit M sera établi sur le premier paquet d'une salve de parole et sur le dernier paquet d'une trame vidéo. Si il était traité comme un champ constant tel que chaque changement exige l'envoi de l'en-tête RTP complet, cela réduirait significativement la compression. Donc, un bit de l'en-tête compressé va porter explicitement le bit M.

Si les paquets s'écoulent à travers un mélangeur RTP, ce qui est très courant pour l'audio, la liste de CSRC et le compte de CC vont aussi changer. Cependant, la liste de CSRC va normalement rester constante durant une salve de parole ou plus longtemps, de sorte qu'il n'est besoin de l'envoyer que lorsque elle change.

### 3.3. Protocole

Le protocole de compression doit tenir une collection d'informations partagées dans un état cohérent entre le compresseur et le décompresseur. Il y a un contexte de session différent pour chaque flux de paquet IP/UDP/RTP, comme défini par une combinaison particulière des adresses IP de source et de destination, les accès UDP de source et de destination, et du champ SSRC RTP. Le nombre de contextes de session à tenir PEUT être négocié entre le compresseur et le décompresseur. Chaque contexte est identifié par un identifiant de 8 ou 16 bits, selon le nombre de contextes négociés, de sorte que le nombre maximum est 65 536. Les paquets non compressés et compressés DOIVENT porter l'identifiant de contexte et un numéro de séquence de quatre bits utilisé pour détecter les pertes de paquet entre le compresseur et le décompresseur. Chaque contexte a son propre espace distinct de numéros de séquence de sorte que la perte d'un seul paquet n'entraîne l'invalidation que d'un seul contexte.

Les informations partagées consistent dans chaque contexte en les éléments suivants :

- o Les en-têtes IP, UDP et RTP complets, incluant éventuellement une liste de CSRC, pour le dernier paquet envoyé par le compresseur ou reconstruit par le décompresseur.
- o La différence de premier ordre pour le champ Identifiant IPv4, initialisé à 1 chaque fois qu'un en-tête IP non compressé pour ce contexte est reçu et est mis à jour chaque fois qu'un delta de champ Identifiant IPv4 est reçu dans un paquet compressé.
- o La différence de premier ordre pour le champ Horodatage RTP, initialisé à 0 chaque fois qu'est reçu un paquet non

compressé pour ce contexte et mis à jour chaque fois qu'est reçu un delta de champ Horodatage RTP dans un paquet compressé.

- o La dernière valeur des quatre bits du numéro de séquence, qui est utilisé pour détecter les pertes de paquet entre le compresseur et le décompresseur.
- o Le numéro de génération actuel pour le codage non différentiel des paquets UDP avec IPv6 (voir la [RFC2507]). Pour IPv4, le numéro de génération peut être réglé à zéro si le type de paquet COMPRESSED\_NON\_TCP, défini ci-dessous, est jamais utilisé.
- o Un tableau des codages de delta spécifiques du contexte (voir au paragraphe 3.3.4) peut facultativement être négocié pour chaque contexte.

Afin de communiquer les paquets dans les diverses formes non compressée et compressée, ce protocole dépend de ce que la couche liaison est capable de donner une indication sur quatre nouveaux formats de paquet en plus des formats normaux de paquet IPv4 et Ipv6 :

**FULL\_HEADER** (*plein en-tête*) – communique l'en-tête IP non compressé plus tous en-têtes et données suivants pour établir l'état d'en-tête non compressé dans le décompresseur pour un certain contexte. Le paquet FULL-HEADER porte aussi l'identifiant de contexte de session de 8 ou 16 bits et le numéro de séquence de quatre bits pour établir la synchronisation entre le compresseur et le décompresseur. Ce format est montré au paragraphe 3.3.1.

**COMPRESSED\_UDP** (*en-tête UDP compressé*) – communique les en-têtes IP et UDP compressés à 6 ou moins d'octets (souvent 2 si les sommes de contrôle UDP sont désactivées) suivis par tous les en-têtes suivants (éventuellement RTP) en forme non compressée, plus les données. Ce type de paquet est utilisé lorsque il y a des différences dans les champs habituellement constants du (potentiel) en-tête RTP. L'en-tête RTP inclut une valeur potentiellement changée du champ SSRC, afin que ce paquet puisse redéfinir le contexte de session. Ce format est montré au paragraphe 3.3.3.

**COMPRESSED\_RTP** (*en-tête RTP compressé*) – indique que l'en-tête RTP est compressé avec les en-têtes IP et UDP. La taille de cet en-tête peut quand même être juste deux octets, ou plus si des différences doivent être communiquées. Ce type de paquet est utilisé lorsque la différence de second ordre (au moins dans les champs usuellement constants) est zéro. Il inclut des codages de delta pour les champs qui ont changé de quantités autres que celles attendues pour établir les différences de premier ordre après l'envoi d'un en-tête RTP non compressé et chaque fois qu'ils changent. Ce format est montré au paragraphe 3.3.2.

**CONTEXT\_STATE** (*état du contexte*) – indique un paquet spécial envoyé du décompresseur au compresseur pour communiquer une liste d'identifiants de contexte pour lesquels la synchronisation a été, ou pourrait être, perdue. Ce paquet n'est envoyé qu'à travers une liaison point à point de sorte qu'il n'exige pas d'en-tête IP. Ce format est donné au paragraphe 3.3.5.

Lorsque ce schéma de compression est utilisé avec IPv6 au titre du cadre général de compression d'en-tête spécifié dans la [RFC2507], un autre type de paquet PEUT être utilisé :

**COMPRESSED\_NON\_TCP** – communique les en-têtes IP et UDP compressés défini dans la [RFC2507] sans codage différentiel. Si il était utilisé pour IPv4, il exigerait un ou deux octets de plus que la forme COMPRESSED\_UDP citée ci-dessus afin de porter le champ Identifiant IPv4. Pour IPv6, il n'y a pas de champ ID et cette compression non différentielle est plus résiliente à la perte de paquet.

L'allocation de codes numériques pour ces formats de paquet dans le protocole point à point [RFC1661] sera faite par l'autorité d'allocation des numéros de l'Internet (IANA).

### 3.3.1 Format de paquet FULL\_HEADER (non compressé)

La définition qu'on donne ici du paquet FULL\_HEADER est destinée à être cohérente avec celle qui est donnée dans la [RFC2507]. Tous les détails sur les choix de conception y sont donnés.

Le format du paquet FULL\_HEADER est le même que celui du paquet original. Dans le cas de IPv4, c'est normalement un en-tête IP, suivi par un en-tête UDP et une charge utile UDP qui peuvent être un en-tête RTP et sa charge utile. Cependant, le paquet FULL\_HEADER peut aussi porter des paquets encapsulés dans IP, auquel cas il va ya avoir deux en-têtes IP suivis par UDP et éventuellement RTP. Ou dans le cas de IPv6, le paquet peut être construit d'une certaine combinaison d'en-têtes IPv6 et IPv4. Chaque en-tête successif est indiqué par le champ Type de l'en-tête précédent, comme d'habitude.

Le paquet FULL\_HEADER diffère du paquet IPv4 ou IPv6 correspondant normal en ce qu'il doit aussi porter l'identifiant de contexte de compression et le numéro de séquence de 4 bits. Afin d'éviter d'augmenter la taille de l'en-tête, ces valeurs sont insérées dans les champs de longueur dans les en-têtes IP et UDP car la longueur réelle peut être déduite de la longueur fournie par la couche liaison. Deux champs Longueur de 16 bits sont nécessaires ; ils sont pris dans les deux premiers en-têtes disponibles dans le paquet. C'est-à-dire, pour un paquet IPv4/UDP, le premier champ de longueur est le champ Longueur totale de l'en-tête IPv4, et le second est le champ Longueur de l'en-tête UDP. Pour un paquet encapsulé en IPv4, le premier champ Longueur viendrait du champ Longueur totale du premier en-tête IP, et le second champ Longueur viendrait du champ Longueur totale du second en-tête IP.

Comme spécifié au paragraphe 5.3.2 de la [RFC2507], la position de l'identifiant de contexte (CID) et du numéro de séquence de quatre bits varie selon que des identifiants de contexte de 8 ou 16 bits ont été choisis, comme le montre le diagramme suivant (large de 16 bits, avec le bit de poids fort à gauche) :

Pour un identifiant de contexte de 8 bits :

```

+-----+
|0|1| Génération|      CID      | premier champ Longueur
+-----+

+-----+
|          0      | seq | second champ Longueur
+-----+

```

Pour un identifiant de contexte de 16 bits :

```

+-----+
|1|1| Génération|  0  | seq | premier champ Longueur
+-----+

+-----+
|          CID      | second champ Longueur
+-----+

```

Le premier bit dans le premier champ Longueur indique la longueur du CID. La longueur du CID DOIT soit être constante pour tous les contextes, soit deux types supplémentaires distincts de paquet DOIVENT être fournis pour indiquer séparément les formats de paquet COMPRESSED\_UDP et COMPRESSED\_RTP avec des CID de 8 et 16 bits. Le second bit dans le premier champ Longueur est 1 pour indiquer que le numéro de séquence de 4 bits est présent, comme c'est toujours le cas pour ce schéma de compression IP/UDP/RTP.

Le champ génération est utilisé avec IPv6 pour les paquets COMPRESSED\_NON\_TCP comme décrit dans la [RFC2507]. Pour les mises en œuvre seulement IPv4 qui n'utilisent pas de paquets COMPRESSED\_NON\_TCP, le compresseur DEVRAIT régler la valeur de génération à zéro. Pour la cohérence du fonctionnement entre IPv4 et IPv6, la valeur de génération est mémorisée dans le contexte lorsque elle est reçue par le décompresseur, et la valeur la plus récente est retournée dans le paquet CONTEXT\_STATE.

Lorsque un paquet FULL\_HEADER est reçu, le jeu complet des en-têtes est mémorisé dans le contexte choisi par l'identifiant de contexte. Le numéro de séquence de 4 bits est aussi mémorisé dans le contexte, resynchronisant ainsi le décompresseur au compresseur.

Lorsque des paquets COMPRESSED\_NON\_TCP sont utilisés, le numéro de séquence de 4 bits est inséré dans le champ "Données" de ce paquet et le bit D est réglé comme décrit à la Section 6 de la [RFC2507]. Lorsque un paquet COMPRESSED\_NON\_TCP est reçu, le numéro de génération est comparé à la valeur mémorisée dans le contexte. Si elles ne sont pas les mêmes, le contexte n'est pas à jour et DOIT être rafraîchi par un paquet FULL\_HEADER. Si la génération correspond, les informations d'en-tête IP et UDP compressé, le numéro de séquence de quatre bits, et l'en-tête (éventuel) RTP sont alors tous mémorisés dans le contexte sauvegardé.

La quantité de mémoire requise pour mémoriser le contexte va varier selon le nombre d'en-têtes d'encapsulation qui sont inclus dans le paquet FULL\_HEADER. Le compresseur et le décompresseur PEUVENT négocier une taille maximum d'en-tête.

### 3.3.2 Format de paquet COMPRESSED\_RTP

Lorsque la différence de second ordre de l'en-tête RTP de paquet à paquet est zéro, le décompresseur peut reconstruire un

paquet simplement en ajoutant les différences de premier ordre mémorisées à l'en-tête non compressé mémorisé représentant le paquet précédent. Tout ce qui doit être communiqué est l'identifiant de contexte de session et un petit numéro de séquence (sans rapport avec le numéro de séquence RTP) pour conserver la synchronisation et détecter les pertes de paquet entre le compresseur et le décompresseur.

Si la différence de second ordre de l'en-tête RTP n'est pas zéro pour certains champs, la nouvelle différence de premier ordre pour ces seuls champs est communiquée en utilisant un codage compact. Les nouvelles valeurs de différence de premier ordre sont ajoutées aux champs correspondants dans l'en-tête non compressé du contexte de session du décompresseur, et sont aussi mémorisées explicitement dans le contexte à ajouter à nouveau aux champs correspondants sur chaque paquet suivant dans lequel la différence de second ordre est zéro. Chaque fois que change la différence de premier ordre, elle est transmise et mémorisée dans le contexte.

En pratique, les seuls champs pour lesquels il est utile de mémoriser la différence de premier ordre sont l'identifiant IPv4 et l'horodatage RTP. Pour le champ Numéro de séquence RTP, l'incrément usuel est 1. Si le numéro de séquence change d'une autre valeur que 1, la différence doit être communiquée mais n'établit pas la différence attendue pour le prochain paquet. La différence de premier ordre attendue reste plutôt fixée à un afin que la différence n'ait pas besoin d'être explicitement communiquée sur le prochain paquet, en supposant qu'il soit à sa place.

Pour l'horodatage RTP, lorsque un paquet FULL\_HEADER, COMPRESSED\_NON\_TCP ou COMPRESSED\_UDP est envoyé pour rafraîchir l'état RTP, la différence de premier ordre mémorisée est initialisée à zéro. Si l'horodatage est le même sur le paquet suivant (par exemple, la même trame vidéo) la différence de second ordre est zéro. Autrement, la différence entre les horodatages des deux paquets est transmise comme nouvelle différence de premier ordre à ajouter à l'horodatage de l'en-tête non compressé mémorisé dans le contexte du décompresseur et aussi mémorisé comme la différence de premier ordre dans ce contexte. Chaque fois que change la différence de premier ordre sur les paquets suivants, cette différence est à nouveau transmise et utilisée pour mettre à jour le contexte.

De façon similaire, comme le champ Identifiant IPv4 s'incrémente fréquemment de un, la différence de premier ordre pour ce champ est initialisée à un lorsque l'état est rafraîchi par un paquet FULL\_HEADER, ou lorsque un paquet COMPRESSED\_NON\_TCP est envoyé car il porte le champ ID sous forme non compressée. À partir de là, chaque fois que la différence de premier ordre change, il est transmis et mémorisé dans le contexte.

On utilisera un gabarit binaire pour indiquer quels champs ont changé d'une différence autre que celle attendue. En plus du petit numéro de séquence de liaison, la liste des éléments à communiquer de façon conditionnelle dans l'en-tête IP/UDP/RTP compressé est la suivante :

I = identifiant de paquet IPv4 (toujours 0 si il n'y a pas d'en-tête IPv4)

U = somme de contrôle UDP

M = bit marqueur de RTP

S = numéro de séquence RTP

T = horodatage RTP

L = compte et liste de CSRC RTP

Si 4 bits sont nécessaires pour que le numéro de séquence de liaison obtienne une probabilité raisonnable de détection de perte, il y a trop peu de bits restants pour allouer un bit à chacun de ces éléments et les faire tenir dans un seul octet avec l'identifiant de contexte.

Il n'est pas nécessaire de porter explicitement le bit U pour indiquer la présence de la somme de contrôle UDP parce qu'une source va normalement inclure des sommes de contrôle sur tous les paquets d'une session ou sur aucun d'entre eux. Lorsque l'état de la session est initialisé avec un en-tête non compressé, si une somme de contrôle différente de zéro est présente, une somme de contrôle de 16 bits non codée sera insérée dans l'en-tête compressé dans tous les paquets suivants jusqu'à ce que le réglage soit changé par l'envoi d'un autre paquet non compressé.

Des éléments restants, le bit L pour le compte et la liste de CSRC est sans doute celui qui est le moins fréquemment utilisé. Plutôt que de dédier un bit du gabarit à indiquer un changement de CSRC, une combinaison inhabituelle des autres bits peut être utilisée à cet effet. Cette combinaison de bits est notée MSTI. Si tous les quatre bits de l'identifiant de paquet IP, du bit marqueur RTP, du numéro de séquence RTP et de l'horodatage RTP sont établis, c'est un cas particulier qui indique qu'une forme étendue de l'en-tête RTP compressé va suivre. Cet en-tête va inclure un octet supplémentaire contenant les valeurs réelles des quatre bits plus le compte CC. La liste CSRC, de longueur indiquée par le compte CC, sera incluse juste comme elle apparaît dans l'en-tête RTP non compressé.

Les autres champs de l'en-tête RTP (version, bit P, bit X, type de charge utile et identifiant de SSRC) sont supposés rester relativement constants. En particulier, l'identifiant de SSRC est défini comme constant pour un certain contexte parce qu'il est un des facteurs qui sélectionnent le contexte. Si un des autres champs change, l'en-tête RTP non compressé DOIT être envoyé comme décrit au paragraphe 3.3.3.

Le diagramme qui suit montre l'en-tête IP/UDP/RTP compressé avec des lignes en pointillé qui indiquent les champs dont la présence est conditionnelle. Le bit de poids fort est numéroté 0. Les champs sur plusieurs octets sont envoyés dans l'ordre des octets du réseau (octet de poids fort en premier). Les champs de delta sont souvent sur un seul octet comme on le voit mais peuvent être sur deux ou trois octets, selon la valeur de delta, comme expliqué au paragraphe 3.3.4.



Quand plus d'un en-tête IPv4 est présent dans le contexte tel qu'initialisé par le paquet FULL\_HEADER, les champs d'identifiants IP des en-têtes encapsulants DOIVENT être envoyés comme valeurs absolues comme décrit dans la [RFC2507]. Ces champs sont identifiés comme champs "ALÉATOIRES". Ils sont insérés dans le paquet COMPRESSED RTP dans le même ordre que celui dans lequel ils apparaissent dans les en-têtes d'origine, suivant immédiatement la somme de contrôle UDP si elle est présente ou l'octet MSTI si elle ne l'est pas, comme le montre le diagramme. C'est seulement si un paquet IPv4 précède immédiatement l'en-tête UDP que l'identifiant IP de cet en-tête sera envoyé de façon différentielle, c'est-à-dire, éventuellement sans aucun bit si la différence seconde est zéro, ou comme un delta de champ Identifiant IPv4 si elle ne l'est pas. Si il n'y a pas un en-tête IPv4 qui précède immédiatement l'en-tête UDP, le bit I DOIT alors être 0 et aucun delta de champ Identifiant IPv4 ne sera présent.

### 3.3.3 Format de paquet COMPRESSED\_UDP

Si il y a un changement dans l'un quelconque des champs de l'en-tête RTP qui sont normalement constants (comme le

champ Type de charge utile) un en-tête RTP non compressé DOIT être envoyé. Si les en-têtes IP et UDP n'ont pas besoin de mise à jour, cet en-tête RTP PEUT être porté dans un paquet COMPRESSED\_UDP plutôt qu'un paquet FULL\_HEADER. Le paquet COMPRESSED\_UDP a le même format que le paquet COMPRESSED\_RTP sauf que les bits M, S et T sont toujours 0 et que les champs delta correspondants ne sont jamais inclus :

```

      0   1   2   3   4   5   6   7
+.....+
:msb d'ID de contexte de session: (si CID de 16 bits)
+-----+
|lsb d'ID de contexte de session|
+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | I | n° sq. liaison |
+---+---+---+---+---+---+---+---+
:                                     :
+   Somme de contrôle UDP           + (si non zéro dans le contexte)
:                                     :
+.....+
:                                     :
+   Champs "ALÉATOIRES"             + (si encapsulés)
:                                     :
+.....+
:   delta identifiant IPv4           : (si I = 1)
+-----+
|                                     |
:   Données UDP                       |
: (en-tête RTP non compressé)       :

```

Noter que ceci constitue une forme de compression d'en-tête IP/UDP différente de celle du type de paquet COMPRESSED\_NON\_TCP définie dans la [RFC2507]. Sa raison est de permettre d'atteindre l'objectif de deux octets lorsque les sommes de contrôle UDP sont désactivées, comme le permet IPv4. Le protocole de la [RFC2507] n'utilise pas le codage différentiel pour les paquets UDP, de sorte que dans le cas de IPv4, deux octets d'identifiant IP, et deux octets de somme de contrôle UDP si elle n'est pas à zéro, seraient toujours transmis en plus des deux octets de préfixe de compression. Pour IPv6, le type de paquet COMPRESSED\_NON\_TCP PEUT être utilisé à la place.

### 3.3.4 Codage des différences

Les champs dans les paquets COMPRESSED\_RTP et COMPRESSED\_UDP sont codés avec une transposition de longueur variable pour des raisons de compacité des valeurs les plus couramment utilisées. On spécifie ci-dessous un codage par défaut, mais il est RECOMMANDÉ que les mises en œuvre utilisent un codeur et décodeur de delta s'appuyant sur un tableau pour permettre la négociation des spécificités du tableau pour chaque session si c'est approprié, et éventuellement même un codage Huffman optimal. Les codages fondés sur l'interprétation séquentielle du flux binaire, dont sont des exemples ce tableau par défaut et le codage Huffman, permettent une taille de tableau raisonnable et peuvent résulter en une vitesse d'exécution supérieure à celle d'une mise en œuvre non fondée sur un tableau avec des essais explicites pour les gammes de valeurs.

Le codage des différences par défaut est spécifié dans le tableau qui suit. Ce codage a été conçu pour coder efficacement les petits changements qui peuvent survenir dans l'identifiant IP et dans le numéro de séquence RTP lorsque des paquets sont perdus en amont du compresseur, tout en traitant la plupart des deltas audio et vidéo sur deux octets. La colonne de gauche est la valeur décimale à coder, et la colonne de droite est la séquence d'octets résultante montrée en hexadécimal et dans l'ordre dans lequel ils sont transmis (ordre des octets du réseau). On montre les premières et les dernières valeurs dans chaque gamme contiguë, avec des points de suspension entre :

Décimal	Hexadécimal
-16384	C0 00 00
:	:
-129	C0 3F 7F
-128	80 00
:	:
-1	80 7F
0	00
:	:
127	7F
128	80 80

:	:
16383	BF FF
16384	C0 40 00
:	:
4194303	FF FF FF

Pour les valeurs positives, un changement de zéro à 127 est représenté directement en un octet. Si les deux bits de poids fort de l'octet sont 10 ou 11, cela signale une extension à une valeur respectivement de deux ou trois octets. Les six bits de moindre poids restants sont combinés, en ordre de poids décroissant avec le ou les deux octets suivants pour former une valeur de 14 ou 22 bits.

Des deltas négatifs surviennent lorsque des paquets sont dérangés ou dans des horodatages RTP intentionnellement déclassés sur de la vidéo MPEG [RFC2250]. Ces événements sont peu probables, de sorte qu'une plus petite gamme de valeurs négatives est codée en utilisant des portions par ailleurs redondantes de la partie positive du tableau.

Un changement de la valeur de l'horodatage RTP de moins de -16 384 ou supérieur à 4 194 303 force l'envoi incompressé de l'en-tête RTP en utilisant un type de paquet FULL\_HEADER, COMPRESSED\_NON\_TCP ou COMPRESSED\_UDP. Les champs Identifiant IP et Numéro de séquence RTP ne font que 16 bits, de sorte que des deltas négatifs pour ces champs DEVRAIENT être masqués à 16 bits et ensuite codés (comme grands nombres positifs de 16 bits).

### 3.3.5 Récupération d'erreur

Chaque fois que le numéro de séquence de 4 bits d'un contexte particulier augmente d'une valeur autre que 1, sauf quand elle est établie par un paquet FULL\_HEADER ou COMPRESSED\_NON\_TCP, le décompresseur DOIT invalider ce contexte et renvoyer un paquet CONTEXT\_STATE au compresseur, indiquant que le contexte a été invalidé. Tous les paquets pour le contexte invalide DOIVENT être éliminés jusqu'à ce que soit reçu un paquet FULL\_HEADER ou COMPRESSED\_NON\_TCP pour ce contexte pour rétablir un état cohérent (sauf si l'algorithme "deux fois" est utilisé comme décrit plus loin dans ce paragraphe). Comme plusieurs paquets compressés peuvent arriver dans l'intervalle, le décompresseur NE DEVRAIT PAS retransmettre de paquet CONTEXT\_STATE pour chaque paquet compressé reçu, mais DEVRAIT plutôt en limiter le taux de retransmission pour éviter d'inonder le canal inverse.

Lorsque survient une erreur sur la liaison, la couche de liaison va normalement éliminer le paquet qui a été endommagé (s'il en est un) mais peut fournir une indication de l'erreur. Un certain temps peut s'écouler avant qu'un autre paquet soit livré pour le même contexte, et alors ce paquet devrait être éliminé par le décompresseur qui va observer qu'il est hors séquence, ce qui résulte en au moins deux paquets perdus. Pour permettre une récupération plus rapide si la liaison fournit une indication d'erreur explicite, le décompresseur PEUT facultativement envoyer un paquet CONTEXT\_STATE pour information faisant la liste du dernier numéro de séquence et du numéro de génération valides pour un ou plusieurs contextes récemment actifs (pas nécessairement tous). Pour un certain contexte, si le compresseur n'avait pas envoyé un paquet compressé avec un numéro de séquence supérieur, et si le numéro de génération correspond à la génération actuelle, aucune action corrective n'est requise. Autrement, le compresseur PEUT choisir de marquer le contexte invalide afin que le prochain paquet soit envoyé en mode FULL\_HEADER ou COMPRESSED\_NON\_TCP (FULL\_HEADER est requis si la génération ne correspond pas). Cependant, noter que si le délai d'aller-retour de la liaison est grand comparé à l'espacement inter paquet, il peut y avoir plusieurs paquets de plusieurs contextes en route à travers la liaison, augmentant la probabilité que les numéros de séquence aient déjà avancé lorsque sera reçu le paquet CONTEXT\_STATE par le compresseur. Le résultat pourrait être que certains contextes sont invalidés sans nécessité, causant une consommation supplémentaire de bande passante.

Le format du paquet CONTEXT\_STATE est montré dans les diagrammes qui suivent. Le premier octet est un code de type pour permettre au type de paquet CONTEXT\_STATE d'être partagé par plusieurs schémas de compression au sein du cadre général de compression spécifié dans la [RFC2507]. Le contenu du reste du paquet dépend du schéma de compression. Pour le schéma de compression IP/UDP/RTP spécifié ici, le reste du paquet CONTEXT\_STATE est structuré comme une liste de blocs pour permettre d'indiquer l'état de plusieurs contextes, précédée par un compte d'un octet du nombre de blocs.

Deux valeurs de code de type sont utilisées pour le schéma de compression IP/UDP/RTP. La valeur 1 indique que des identifiants de contexte de session de 8 bits sont utilisés :

```

  0  1  2  3  4  5  6  7
+---+---+---+---+---+---+---+---+
|1 = IP/UDP/RTP avec CID 8 bits |
+---+---+---+---+---+---+---+---+
|           compte de contexte           |
+---+---+---+---+---+---+---+---+
|           ID de contexte de session     |
+---+---+---+---+---+---+---+---+
| I | 0 | 0 | 0 | n° sequence |
+---+---+---+---+---+---+---+---+
| 0 | 0 | n° de génération |
+---+---+---+---+---+---+---+---+
...
+---+---+---+---+---+---+---+---+
|           ID de contexte de session     |
+---+---+---+---+---+---+---+---+
| I | 0 | 0 | 0 | n° sequence |
+---+---+---+---+---+---+---+---+
| 0 | 0 | n° de génération |
+---+---+---+---+---+---+---+---+

```

La valeur 2 indique que des identifiants de contexte de session de 16 bits sont utilisés. L'identifiant de contexte de session est envoyé dans l'ordre des octets du réseau (bit de poids fort en premier) :

```

  0  1  2  3  4  5  6  7
+---+---+---+---+---+---+---+---+
|2 = IP/UDP/RTP avec CID 16 bits|
+---+---+---+---+---+---+---+---+
|           Compte de contexte           |
+---+---+---+---+---+---+---+---+
|           Identifiant contexte de session+
+---+---+---+---+---+---+---+---+
| I | 0 | 0 | 0 | n° de séquence|
+---+---+---+---+---+---+---+---+
| 0 | 0 | n° de génération |
+---+---+---+---+---+---+---+---+
...
+---+---+---+---+---+---+---+---+
|           ID de contexte de session     +
+---+---+---+---+---+---+---+---+
| I | 0 | 0 | 0 | n° de séquence|
+---+---+---+---+---+---+---+---+
| 0 | 0 | n° de génération |
+---+---+---+---+---+---+---+---+

```

Le bit marqué "I" est réglé à un pour les contextes qui ont été marqués invalides et requièrent que soit transmis un FULL\_HEADER de paquet COMPRESSED\_NON\_TCP. Si le bit I est zéro, l'état du contexte est pour information. Le bit I est réglé à zéro pour indiquer l'état de contexte indicatif qui PEUT être envoyé suite à une indication d'erreur de liaison.

Comme le paquet CONTEXT\_STATE lui-même peut être perdu, la retransmission d'un ou de plusieurs blocs est permise. On s'attend à ce que la retransmission ne soit déclenchée que par la réception d'un autre paquet, mais si la ligne est presque oisive, la retransmission PEUT être déclenchée par un temporisateur relativement long (de l'ordre de 1 seconde).

Si un bloc de CONTEXT\_STATE est retransmis pour un certain contexte, il peut croiser des chemins avec le paquet FULL\_HEADER ou COMPRESSED\_NON\_TCP destiné à rafraîchir ce contexte. Dans ce cas, le compresseur PEUT choisir d'ignorer l'indication d'erreur.

Dans le cas où des sommes de contrôle UDP sont transmises, le décompresseur PEUT tenter d'utiliser l'algorithme "deux fois" décrit au paragraphe 10.1 de la [RFC2507]. Dans cet algorithme, le delta est appliqué plus d'une fois avec l'hypothèse que le delta peut avoir été le même sur le ou les paquets manquants que sur celui reçu ensuite. La réussite est indiquée par la correspondance de la somme de contrôle. Pour le schéma défini ici, la différence du numéro de séquence de 4 bits dit le nombre de fois que le delta doit être appliqué. Noter cependant qu'il y a un risque non trivial d'une indication positive incorrecte. Il peut être conseillé de demander un paquet FULL\_HEADER ou COMPRESSED\_NON\_TCP même si l'algorithme "deux fois" réussit.

Certaines erreurs peuvent n'être pas détectées, par exemple si 16 paquets de suite sont perdus et si le niveau liaison ne donne pas d'indication d'erreur. Dans ce cas, le décompresseur va générer des paquets qui ne sont pas valides. Si les sommes de contrôle UDP sont transmises, le receveur va probablement détecter les paquets invalides et les éliminer, mais le receveur n'a aucun moyen de le signaler au décompresseur. Donc, il est RECOMMANDÉ que le décompresseur vérifie périodiquement la somme de contrôle UDP, peut-être une fois tous les 16 paquets. Si une erreur est détectée, le décompresseur va invalider le contexte et le signaler au compresseur par un paquet CONTEXT\_STATE.

### 3.4 Compression des paquets de contrôle RTCP

En s'appuyant sur la convention RTP que les données sont portées sur un numéro d'accès pair et que les paquets RTCP correspondants sont portés sur le numéro d'accès supérieur (impair) on pourrait dessiner des schémas de compression distincts à appliquer aux paquets RTP et RTCP. Pour RTCP, la compression pourrait s'appliquer non seulement à l'en-tête mais aussi aux "données", c'est-à-dire, au contenu des différents types de paquets. Les nombres des paquets RTCP de rapport d'envoyeur (SR, *Sender Report*) et de rapport de receveur (RR, *Receiver Report*) ne se compresseraient pas bien, mais les informations de texte dans les paquets de description de source (SDS, *Source Description*) pourraient être compressés jusqu'à un gabarit binaire indiquant chaque élément qui était présent mais a été compressé (pour des besoins de synchronisation sur l'élément NOTE de SDS et pour permettre au système d'extrémité de mesurer la taille moyenne de paquet RTCP pour le calcul d'intervalle).

Cependant, dans le schéma de compression défini ici, aucune compression ne sera faite sur les en-têtes et "données" RTCP pour plusieurs raisons (mais la compression DEVRAIT quand même être appliquée aux en-têtes IP et UDP). Comme la spécification du protocole RTP suggère que l'intervalle du paquet RTCP soit adaptée de telle sorte que la bande passante RTCP agrégée utilisée par tous les participants à une session ne représente pas plus de 5 % de la bande passante de session, il n'y a pas grand chose à gagner de la compression RTCP. Compresser les éléments de SDS exigerait un accroissement significatif de l'état partagé qui devrait être mémorisé pour chaque identifiant de contexte. Et, afin de permettre la compression lorsque des informations de SDS pour plusieurs sources ont été envoyées à travers un "mélangeur" RTP, il serait nécessaire de tenir un contexte de session RTCP séparé pour chaque identifiant de SSRC. Dans une session avec plus de 255 participants, cela causerait un bourrage complet de l'antémémoire du contexte même lorsque un seul participant envoie des données.

Même si RTCP n'est pas compressé, la fraction de la bande passante totale occupée par les paquets RTCP sur la liaison compressée reste inférieure à 5 % dans la plupart des cas, en supposant que les paquets RTCP sont envoyés comme paquets COMPRESSED\_UDP. Étant donné que le trafic RTCP non compressé ne consomme pas plus de 5 % de la bande passante totale de la session, pour une longueur normale de paquet RTCP de 90 octets, la portion de la bande passante compressée utilisée par RTCP ne fera pas plus de 5 % si la taille de la charge utile dans les paquets de données RTP est au moins de 108 octets. Si la taille de la charge utile de données RTP est plus petite, cette fraction va augmenter, mais reste inférieure à 7 % pour une taille de charge utile de 37 octets. Pour les grosses charges utiles de données, la fraction RTCP compressée est inférieure à la fraction RTCP non compressée (par exemple, 4 % à 1000 octets).

### 3.5 Compression de paquets UDP non RTP

Comme décrit précédemment, le paquet COMPRESSED\_UDP PEUT être utilisé pour compresser des paquets UDP qui ne portent pas RTP. Quelles que soient les données qui suivent, il est peu probable que l'en-tête UDP ait des valeurs constantes dans les bits qui correspondent aux champs habituellement constants dans l'en-tête RTP. En particulier, le champ SSRC va vraisemblablement changer. Il est donc nécessaire de garder trace des flux de paquets UDP non RTP pour éviter de consommer tous les intervalles de contexte lorsque change le "champ SSRC" (car ce champ fait partie de ce qui identifie un contexte RTP particulier). Ces flux peuvent recevoir chacun un contexte, mais le codeur va établir un fanion dans le contexte pour indiquer que le champ SSRC qui change devrait être ignoré et que les paquets COMPRESSED\_UDP devraient toujours être envoyés plutôt que des paquets COMPRESSED\_RTP.

## 4. Interaction avec la segmentation

Un schéma de segmentation peut être utilisé en conjonction avec la compression d'en-tête RTP pour permettre que des petits paquets en temps réel interrompent des gros paquets, supposés n'être pas en temps réel, afin de réduire le délai. On suppose que les gros paquets outrepassent le compresseur et le décompresseur car l'entrelacement modifierait le séquençage des paquets chez le décompresseur et causerait l'apparition d'erreurs. La compression d'en-tête devrait être moins importantes pour les gros paquets car le taux de redondance est plus faible.

Si certains paquets provenant d'un contexte de session RTP sont choisis pour une segmentation (peut-être sur la base de la taille) et d'autres ne le sont pas, il y a une possibilité de réarrangement. Cela réduirait l'efficacité de compression parce que les gros paquets vont apparaître comme des paquets perdus dans la suite des numéros de séquence. Cependant, cela ne devrait pas poser de problèmes plus sérieux parce que les numéros de séquence RTP devraient être reconstruits correctement et vont permettre à l'application de corriger l'ordre.

Les erreurs de liaison détectées par le schéma de segmentation qui utilise ses propres informations de séquençage PEUVENT être indiquées au compresseur par un message CONTEXT\_STATE d'information tout comme pour les erreurs de liaison détectées par la couche de liaison elle-même.

L'octet Identifiant de contexte est placé en premier dans l'en-tête COMPRESSED\_RTP de sorte que cet octet PEUT être partagé avec la couche de segmentation si un tel partage est faisable et a été négocié. Comme le compresseur peut allouer des valeurs arbitraires d'identifiant de contexte, la valeur peut être réglée de façon à correspondre à l'identifiant de contexte de la couche de segmentation.

## 5. Négociation de la compression

L'utilisation de la compression IP/UDP/RTP sur une certaine liaison est une fonction du protocole de couche liaison. On suppose qu'une telle négociation sera définie séparément pour PPP [RFC1661], par exemple. Les éléments suivants PEUVENT être négociés :

- o la taille de l'identifiant de contexte,
- o la taille maximum de la pile d'en-têtes dans le contexte,
- o un tableau spécifique du contexte pour décoder les valeurs de deltas.

## 6. Remerciements

Plusieurs personnes ont contribué à la conception de ce schéma de compression et des problèmes qui s'y rapportent. Scott Petrack a initié la discussion de la compression de l'en-tête RTP dans le groupe de travail AVT à Los Angeles en mars 1996. Carsten Bormann a développé une architecture globale pour la compression en combinaison avec le contrôle du trafic sur une liaison à bas débit, et fait plusieurs contributions spécifiques au schéma décrit ici. David Oran a développé indépendamment une note fondée sur des idées similaires, et a suggéré l'utilisation du protocole PPP multi liaison pour la segmentation. Mikael Degermark a contribué par son avis sur l'intégration de ce schéma de compression dans le cadre de la compression IPv6.

## 7. Références

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

[RFC1661] W. Simpson, éditeur, "[Protocole point à point](#) (PPP)", STD 51, juillet 1994. (MàJ par la RFC2153)

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : protocole de transport pour applications en temps réel", janvier 1996. (Obsolète, voir [RFC3550 STD64](#))

[RFC2250] D. Hoffman et autres, "[Format de charge utile RTP](#) pour la vidéo MPEG1/MPEG2", janvier 1998.

[RFC2507] M. Degermark, B. Nordgren, S. Pink, "[Compression d'en-tête IP](#)", février 1999. (P.S.)

## 8. Considérations pour la sécurité

Comme le chiffrement élimine la redondance que ce schéma de compression essaye d'exploiter, on peut être incité à ne pas faire de chiffrement afin de réaliser le fonctionnement sur une liaison à faible bande passante. Cependant, pour les cas où le chiffrement des données et pas des en-têtes est satisfaisant, RTP spécifie une méthode de chiffrement de remplacement dans laquelle seule la charge utile RTP est chiffrée et les en-têtes sont laissés en clair. Cela permettrait que la compression soit quand même appliquée.

Un compresseur défaillant ou malveillant pourrait être cause que le décompresseur reconstitue des paquets qui ne correspondent pas aux paquets d'origine mais ont quand même des en-têtes IP, UDP et RTP valides et éventuellement même des sommes de contrôle UDP valides. Une telle corruption peut être détectée avec des mécanismes d'authentification et de protection de l'intégrité de bout en bout qui ne seront pas affectés par la compression. Les portions constantes des en-têtes d'authentification seront compressées comme décrit dans la [RFC2507].

Aucune authentification n'est effectuée sur le paquet de contrôle CONTEXT\_STATE envoyé par le présent protocole. Un attaquant qui a accès à la liaison entre le décompresseur et le compresseur pourrait injecter de faux paquets CONTEXT\_STATE et causer une réduction de l'efficacité de compression, résultant probablement en encombrement sur la liaison. Cependant, un attaquant qui a accès à la liaison pourrait aussi interrompre le trafic de nombreuses autres façons.

Il existe une menace potentielle de déni de service lorsque on utilise des techniques de compression qui ont une charge de calcul non uniforme à l'extrémité de réception. L'attaquant peut injecter des datagrammes pathologiques dans les flux qui sont complexes à décompresser et causer la surcharge du receveur et la dégradation du traitement des autres flux. Cependant, la présente compression ne présente aucune non uniformité significative.

Un passage en revue de la sécurité du présent protocole n'a décelé aucune considération de sécurité supplémentaire.

## 9. Adresse des auteurs

Stephen L. Casner  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
United States  
mél : [casner@cisco.com](mailto:casner@cisco.com)

Van Jacobson  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
United States  
mél : [van@cisco.com](mailto:van@cisco.com)

## 10. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1999). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de droits de reproduction ci-dessus et le présent paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou ses successeurs ou ayant droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

### Remerciement

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