

Groupe de travail Réseau
Request for Comments : 4815
 RFC mises à jour : 3095, 3241, 3843, 4019, 4362
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

L-E. Jonsson
 K. Sandlund
 G. Pelletier
 P. Kremer
 février 2007

Compression d'en-tête robuste (ROHC) : corrections et précisions à la RFC 3095

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir 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 (2007).

Résumé

La RFC 3095 définit le cadre de la compression d'en-tête robuste (ROHC, *RObust Header Compression*) et ses profils pour le protocole Internet (IP, *Internet Protocol*) le protocole de datagramme d'utilisateur (UDP, *User Datagram Protocol*) le protocole de transport en temps réel (RTP, *Real-Time Transport Protocol*) et la charge utile de sécurité encapsulante (ESP, *Encapsulating Security Payload*). Certaines parties de la spécification ne sont pas claires ou contiennent des erreurs qui peuvent conduire à de mauvaises interprétations qui pourraient nuire à l'interopérabilité entre des mises en œuvre différentes. Le présent document fournit des corrections, des ajouts, et des éclaircissements à la RFC 3095 ; il met donc à jour la RFC 3095. De plus, d'autres éclaircissements relatifs aux RFC 3241 (ROHC sur PPP) RFC 3843 (profil ROHC pour IP) et RFC 4109 (profils ROHC pour UDP léger) sont aussi fournis.

Table des matières

1. Introduction et terminologie.....	2
2. Calcul et couverture du CRC	3
2.1 Calcul du CRC	3
2.2. Octet de bourrage et calculs de CRC.....	3
2.3 Couverture du CRC dans les options de retour de CRC.....	3
2.4 Couverture par le CRC de l'en-tête ESP NULL.....	3
3. Transition de mode.....	3
3.1 Retours durant la transition de mode U et de mode O.....	3
3.2 Rétroaction durant une transition de mode.....	5
3.3 Décodage de paquet durant une transition de mode.....	5
4. Codage d'horodatage.....	6
4.1 Codage utilisé pour les bits TS compressés.....	6
4.2 (Dé)compression de TS sans bits TS transmis.....	6
4.3 Intervalles d'interprétation pour le codage d'horodatage.....	7
4.4 Codage d'horodatage RTP adapté.....	7
4.5 Recalcul de TS_OFFSET.....	8
4.6 TS_STRIDE et le fanion Tsc dans l'extension 3.....	8
4.7 Utilisation de la compression fondée sur un temporisateur.....	9
5. Compression de liste.....	9
5.1 Éléments de liste de CSRC dans la chaîne dynamique RTP.....	9
5.2 Occurrences multiples du champ CC.....	9
5.3 Gabarits binaires dans une compression de liste.....	10
5.4 En-têtes compressés avec compression de liste.....	10
5.5 Compression de liste d'en-têtes ESP NULL.....	10
5.6 Tableaux et indices de traduction pour les en-têtes d'extension IP.....	10
5.7 Liste de références.....	10
5.8. Compression de numéros de séquence AH et GRE.....	11
6. Mise à jour de propriétés.....	11

6.1 Mises à jour implicites.....	11
6.2 Mise à jour des propriétés de UO-1*.....	12
6.3 Propriétés de mise à jour de contexte pour paquets IR.....	12
6.4 Champ de bourrage RTP (R-P) dans l'extension 3.....	12
6.5 Bit eXtension (X) RTP dans la partie dynamique.....	12
7. Gestion de contexte et réutilisation de CID/contexte.....	13
7.1 Persistance de contextes de décompresseur.....	13
7.2 Réutilisation de CID/contexte.....	13
8. Autres précisions au protocole.....	14
8.1 Signification de NBO.....	14
8.2 IP-ID.....	14
8.3 Extension 3 dans les paquets UOR-2*.....	14
8.4 Occurrences multiples du bit M.....	14
8.5 Options SN multiples dans un paquet de rétroaction.....	15
8.6 Options de CRC multiples dans un paquet de rétroaction.....	15
8.7 Réponse à la perte de liaisons de rétroaction.....	15
8.8 UOR-2 dans le profil 0x0002 (UDP) et le profil 0x0003 (ESP).....	15
8.9 Bits de moindre poids du numéro de séquences dans les en-têtes d'extension IP.....	15
8.10 Attente d'accusé de réception UOR-2 en mode O.....	15
8.11 Réparations de contexte, TS_STRIDE et TIME_STRIDE.....	16
9. Négociation de ROHC.....	16
10. Sous-option PROFILES dans ROHC sur PPP.....	16
11. Codage IP-ID constant dans les profils IP seul et UPD léger.....	16
12. Considérations sur la sécurité.....	17
13. Remerciements.....	17
14. Références.....	17
14.1 Références normatives.....	17
14.2 Références pour information.....	17
Appendice A. Exemple d'algorithme de CRC.....	18
Adresse des auteurs.....	20
Déclaration complète de droits de reproduction.....	20

1. Introduction et terminologie

La [RFC3095] définit le cadre de la compression d'en-tête robuste (ROHC, *RObust Header Compression*) et ses profils pour IP [RFC0791], [RFC2460], UDP [RFC0768], RTP [RFC3550], et ESP [RFC4303]. Durant les essais de mise en œuvre et d'interopérabilité de la RFC 3095, certaines ambiguïtés et mauvaises interprétations courantes ont été identifiées, ainsi que quelques erreurs.

Le présent document résume les problèmes identifiés et donne des corrections nécessaires pour que les mises en œuvre de la RFC 3095 interopèrent, c'est-à-dire qu'il constitue une mise à jour de la RFC 3095. Le présent document fournit aussi d'autres éclaircissements relatifs à des mauvaises interprétations courantes de la spécification. Les références à la RFC3095 devraient donc aussi inclure le présent document.

De plus, des éclaircissements et corrections sont aussi fournis pour la [RFC3241] (ROHC sur PPP), la [RFC3843] (profil ROHC IP seul) et la [RFC4019] (profils ROHC UDP léger) qui sont donc aussi mises à jour par le présent document. De plus, la [RFC4362] (profil ROHC assisté de couche de liaison) est implicitement mise à jour par le présent document, car la RFC 4362 est aussi fondée sur la RFC 3095.

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

Quand un paragraphe de ce document fait des corrections, ajouts ou invalidations formelles au texte de la RFC 3095, cela est clairement mentionné. Le texte de la RFC 3095 qui est visé est donné et est marqué "INCOMPLET", "INCORRECT", ou "INCORRECT ET INVALIDÉ", suivi par le texte correct marqué "CORRIGÉ", lorsque applicable. Quand du texte est ajouté sans qu'il corrige simplement le texte des spécifications antérieures, il est donné avec l'étiquette "AJOUT FORMEL".

Dans ce document, une référence à un paragraphe de la [RFC3095] est marqué RFC 3095, paragraphe <numéro>.

2. Calcul et couverture du CRC

2.1 Calcul du CRC

La RFC 3095, paragraphe 5.9 définit des polynômes pour des contrôles de redondance cycliques (CRC, *Cyclic Redundancy Check*) de 3, 7, et 8 bits, mais elle ne spécifie pas l'algorithme utilisé. Les CRC de 3, 7, et 8 bits sont calculés avec l'algorithme de CRC défini dans la [RFC1662].

Une mise en œuvre Perl de l'algorithme se trouve à l'Appendice A du présent document.

2.2. Octet de bourrage et calculs de CRC

La RFC 3095, paragraphe 5.9.1 est incomplète, car elle ne mentionne pas comment traiter l'octet de bourrage dans les calculs de CRC pour les paquets IR et IR-DYN. Le bourrage n'est pas destiné à être une partie signifiante d'un paquet et n'est pas inclus dans le calcul de CRC. Par suite, le CRC ne couvre pas non plus l'octet Add-CID pour CID 0.

TEXTE INCOMPLET de la RFC 3095, paragraphe 5.9.1 :

"Le CRC dans le paquet IR et IR-DYN est calculé sur le paquet IR ou IR-DYN entier, en excluant la charge utile et en incluant le CID ou tout octet Add-CID."

TEXTE CORRIGÉ :

"Le CRC dans le paquet IR et IR-DYN est calculé sur le paquet IR ou IR-DYN entier, en excluant la charge utile, le bourrage, et en incluant le CID ou tout octet Add-CID, sauf l'octet add-CID pour CID 0."

2.3 Couverture du CRC dans les options de retour de CRC

La RFC 3095, paragraphe 5.7.6.3 est incomplète, car elle ne mentionne pas comment est traité le champ "Taille" quand on calcule le CRC de 8 bits utilisé dans l'option Retour de CRC. Comme le champ "Taille" est une extension du champ "Code", il doit être traité de la même façon.

TEXTE INCOMPLET de la RFC 3095, paragraphe 5.7.6.3 :

"L'option CRC contient un CRC de 8 bits calculé sur la charge utile entière de retour, sans le type de paquet et l'octet de code, mais incluant tous les champs de CID, en utilisant le polynôme du paragraphe 5.9.1."

TEXTE CORRIGÉ :

"L'option CRC contient un CRC de 8 bits calculé sur la charge utile entière de retour incluant tous les champs de CID mais excluant le type de paquet, le champ Taille et l'octet Code, en utilisant le polynôme du paragraphe 5.9.1."

2.4 Couverture par le CRC de l'en-tête ESP NULL

La RFC 3095, paragraphe 5.8.7 donne la couverture du CRC de l'en-tête ESP NULL [RFC2410] comme "en-tête ESP entier". Cela doit être interprété comme incluant seulement la partie initiale de l'en-tête (c'est-à-dire, l'indice de paramètre de sécurité (SPI, *Security Parameter Index*) et le numéro de séquence) et non la partie en-queue à la fin de la charge utile. Donc, l'en-tête ESP NULL a la même couverture de CRC que l'en-tête ESP utilisé dans le profil ESP (RFC 3095, paragraphe 5.7.7.7).

3. Transition de mode

3.1 Retours durant la transition de mode U et de mode O

La RFC 3095, paragraphe 5.6.1 déclare que durant les transitions de mode, lorsque le paramètre D_TRANS est I, le décompresseur envoie des retours pour chaque paquet reçu. Ce comportement restrictif empêche le décompresseur d'utiliser un algorithme de retours éparés durant les transitions de mode.

Pour réduire les frais généraux de transmission et la complexité du calcul (incluant le calcul de CRC) associés aux paquets

le décompresseur accuse réception d'un (ou plusieurs) paquets IR, IR-DYN ou UOR-2 qui ont les bits de mode réglés au nouveau mode. Les paquets de type 0 ou de type 1 qui sont reçus jusqu'à ce point sont décompressés en utilisant le vieux mode, tandis qu'ensuite, ils sont décompressés en utilisant le nouveau mode. Si les procédures de transition améliorées décrites au paragraphe 3.1 sont utilisées, le réglage du paramètre D_TRANS à P représente ce point de rupture. La réussite de la décompression d'un paquet de type 0 ou de type 1 achève la transition de mode.

4. Codage d'horodatage

4.1 Codage utilisé pour les bits TS compressés

Les valeurs d'horodatage (TS, *Timestamp*) RTP sont toujours codées en utilisant le codage W-LSB, envoyées aussi bien adaptées que non adaptées. Quand aucun bit TS n'est transmis dans un paquet compressé, TS est toujours adapté. Si un paquet compressé porte une Extension 3 et un champ (Tsc)=0, le paquet compressé doit donc toujours porter des bits TS non adaptés. Pour les valeurs de TS envoyées dans Extension 3, les valeurs codées en W-LSB sont envoyées en utilisant le format auto descriptif de longueur variable (RFC3095, paragraphe 4.5.6) et cela s'applique aux valeurs adaptées et non adaptées.

4.2 (Dé)compression de TS sans bits TS transmis

Quand ROHC RTP opère en utilisant ses types de paquets les plus efficaces, à part l'identification de type de paquet et le CRC de détection d'erreur, seuls les bits de numéro de séquence (SN) RTP sont transmis dans les en-têtes RTP compressés. Tous les autres champs sont alors omis soit parce que ils sont inchangés, soit parce que ils peuvent être reconstruits par une fonction du SN (c'est-à-dire, en combinant les bits de SN transmis avec les informations d'état provenant du contexte). Les champs qui peuvent être déduits du SN sont l'identification IP (IP-ID) et l'horodatage RTP (TS).

La compression et la décompression de IP-ID, avec et sans bits IP-ID transmis dans l'en-tête compressé, sont bien définies dans la RFC3095, paragraphe 4.5.5 (voir au paragraphe 8.2). Pour le champ TS, la RFC 3095 définit comment décompresser sur la base des bits TS réels dans l'en-tête compressé, adapté ou non adapté, mais pas comment déduire le TS du SN quand il n'y a pas de bit TS présent dans l'en-tête compressé.

Quand aucun bit TS n'est reçu dans l'en-tête compressé, la valeur de TS adaptée est reconstruite en supposant une extrapolation linéaire du SN, c'est-à-dire, $\text{delta_TS} = \text{delta_SN} * \text{pente}$ par défaut, où delta_SN et delta_TS sont tous deux des entiers signés. La RFC3095, paragraphe 5.7 définit les valeurs potentielles pour la pente par défaut.

TEXTE INCOMPLET de la RFC 3095, paragraphe 5.7 :

Si $\text{valeur}(\text{Tsc}) = 1$, on utilise le codage d'horodatage RTP adapté avant la compression (voir au paragraphe 4.5.3), et pente par défaut(TS) = 1.

Si $\text{valeur}(\text{Tsc}) = 0$, la valeur de l'horodatage est compressée telle qu'elle, et pente par défaut(TS) = $\text{valeur}(\text{TS_STRIDE})$.

TEXTE CORRIGÉ :

"Quand un en-tête compressé est reçu sans bit TS, la valeur de TS adaptée est reconstruite en supposant une extrapolation linéaire du SN, c'est-à-dire, $\text{delta_TS} = \text{delta_SN} * \text{pente}$ par défaut(TS).

Si $\text{valeur}(\text{Tsc}) = 1$, on utilise le codage d'horodatage RTP adapté avant la compression (voir au paragraphe 4.5.3), et pente par défaut(TS) = 1.

Si $\text{valeur}(\text{Tsc}) = 0$, la valeur de l'horodatage est compressée telle qu'elle, et pente par défaut(TS) = $\text{valeur}(\text{TS_STRIDE})$. Si un paquet sans bit TS est reçu avec $\text{Tsc} = 0$, le décompresseur DOIT éliminer le paquet."

TEXTE INCORRECT ET INVALIDÉ DE LA RFC 3095 paragraphe 5.5.1.2):

"Par exemple, dans un cas normal où le schéma de la chaîne a la forme d'un champ non SN = SN * pente + décalage, un ACK est suffisant si la pente a été précédemment établie par le décompresseur (c'est-à-dire, seul le nouveau décalage a besoin d'être synchronisé). Autrement, deux ACK sont nécessaires car le décompresseur a besoin de deux en-têtes pour apprendre à la fois la nouvelle pente et le nouveau décalage."

Par conséquent, il n'y a pas d'autre valeur de pente que la pente par défaut, commé défini dans la RFC3095, paragraphe 5.7.

4.3 Intervalles d'interprétation pour le codage d'horodatage

La RFC 3095, paragraphe 4.5.4 définit l'intervalle d'interprétation, p , pour la compression fondée sur un temporisateur de l'horodatage RTP. Cependant, la RFC 3095, paragraphe 5.7 définit un intervalle d'interprétation différent, comme l'intervalle d'interprétation à utiliser pour toutes les valeurs d'horodatage. La valeur de p à utiliser n'est donc pas claire, au moins en ce qui concerne la compression fondée sur un temporisateur.

La façon dont ceci devrait être interprété est que la valeur de p diffère selon que la compression fondée sur un temporisateur est activée ou non.

Pour la compression fondée sur un temporisateur (TIME_STRIDE réglé à une valeur non zéro), l'intervalle d'interprétation est :

$$p = 2^{(k-1)} - 1 \text{ (conformément à la RFC3095, paragraphe 4.5.4)}$$

Autrement, l'intervalle d'interprétation est :

$$p = 2^{(k-2)} - 1 \text{ (conformément à la RFC3095, paragraphe 5.7)}$$

4.4 Codage d'horodatage RTP adapté

Ce paragraphe redéfinit l'algorithme pour le codage d'horodatage RTP adapté, défini comme une procédure en cinq étapes dans la RFC3095, paragraphe 4.5.3. Deux erreurs formelles ont été corrigées, comme décrit dans les sous paragraphes 4.4.1 et 4.4.2, et l'algorithme entier a été revu pour être plus concis et utiliser une terminologie bien définie. Le texte résultant se trouve en 4.4.3.

4.4.1 TS_STRIDE pour codage d'horodatage adapté

La RFC 3095 définit le pas d'horodatage (TS_STRIDE) comme l'augmentation attendue de la valeur de l'horodatage entre deux paquets RTP avec des numéros de séquence consécutifs. TS_STRIDE est réglé par le compresseur et explicitement communiqué au décompresseur, et il est utilisé comme facteur d'adaptation pour le codage de TS adapté.

La relation entre TS et TS_SCALED, donnée par l'égalité suivante de la RFC3095, paragraphe 4.5.3, définit la signification mathématique de TS_STRIDE :

$$TS = TS_SCALED * TS_STRIDE + TS_OFFSET$$

TS_SCALED est incomplètement décrit comme TS / TS_STRIDE dans l'étape de compression qui suit l'égalité centrale ci-dessus. Cette formule est incorrecte à la fois parce que elle exclut TS_OFFSET et parce que elle empêcherait une valeur de TS_STRIDE de 0, ce qui est une alternative non exclue par la définition ou par l'égalité centrale ci-dessus. Si "/" était une opération généralement définie de façon non ambiguë signifiant "la partie entière du résultat de la division de X par Y", l'absence de TS_OFFSET pourrait être expliquée, mais la formule manquerait quand même d'un résultat approprié pour $TS_STRIDE = 0$. La formule de "2. Compression" n'est donc valide qu'avec les exigences suivantes :

- a) "/" signifie "la partie entière du résultat de la division de X par Y"
- b) $TS_STRIDE > 0$ (TS n'est jamais envoyé adapté quand $TS_STRIDE = 0$)

4.4.2 Retour à zéro d'horodatage avec codage d'horodatage adapté

La RFC 3095, paragraphe 4.5.3 déclare aux points 4 et 5 que le compresseur n'est pas obligé d'initialiser TS_OFFSET au retour à zéro, mais qu'il est obligé d'augmenter le nombre de bits envoyés pour la valeur de TS adaptée quand il y a un retour à zéro de l'horodatage. Le décompresseur est aussi obligé de détecter et de s'accomoder du retour à zéro de l'horodatage, incluant de mettre à jour TS_OFFSET.

Cette méthode n'est pas interopérable et n'est pas robuste. Le gain est aussi insignifiant, car le retour à zéro de l'horodatage n'arrive que très rarement. Donc, le compresseur devrait réinitialiser TS_OFFSET lors d'un retour à zéro de l'horodatage, en envoyant un TS non adapté.

4.4.3 Algorithme pour codage d'horodatage adapté

TEXTE INCORRECT ET INVALIDÉ DE LA RFC 3095 paragraphe 4.5.3):

1. Initialisation : Le compresseur envoie au décompresseur la valeur de TS_STRIDE et la valeur absolue d'un ou plusieurs champs TS. Ces dernières sont utilisées par le décompresseur pour initialiser TS_OFFSET à (valeur absolue) modulo TS_STRIDE. Noter que TS_OFFSET est le même sans considération de la valeur absolue utilisée, tant que la valeur TS non adaptée ne revient pas à zéro ; voir le 4) ci-dessous.
2. Compression : Après l'initialisation, le compresseur ne compresse plus les valeurs de TS d'origine. Il compresse plutôt les valeurs diminuées : $TS_SCALED = TS / TS_STRIDE$. La méthode de compression pourrait être le codage W-LSB ou le codage fondé sur le temporisateur décrit au paragraphe suivant.
3. Décompression : Lorsque il reçoit la valeur compressée de TS_SCALED, le décompresseur déduit d'abord la valeur du TS_SCALED original. Le TS RTP original est alors calculé comme $TS = TS_SCALED * TS_STRIDE + TS_OFFSET$.
4. Décalage au retour à zéro : Le retour à zéro du TS de 32 bits non adapté va invalider la valeur courante de TS_OFFSET utilisée dans l'équation ci-dessus. Par exemple, supposons que $TS_STRIDE = 160 = 0xA0$ et que le TS courant = $0xFFFFFFFF0$. TS_OFFSET est alors de $0x50 = 80$. Si le prochain TS RTP = $0x00000130$ (c'est-à-dire, si l'incrément est $160 * 2 = 320$) le nouveau TS_OFFSET devrait être $0x00000130$ modulo $0xA0 = 0x90 = 144$. Le compresseur n'est pas obligé de réinitialiser TS_OFFSET au moment du retour à zéro. Le décompresseur DOIT plutôt détecter le retour à zéro du TS non adapté (ce qui est trivial) et mettre à jour TS_OFFSET à $TS_OFFSET = (TS \text{ non adapté revenu à zéro}) \text{ modulo } TS_STRIDE$.

TEXTE CORRIGÉ :

- "1. Initialisation et mise à jour de la fonction d'adaptation de l'horodatage RTP : le compresseur envoie au décompresseur la valeur de TS_STRIDE ainsi qu'un horodatage non adapté. Elles sont toutes deux nécessaires au décompresseur pour initialiser TS_OFFSET à En-tête(TS) modulo champ(TS_STRIDE). Noter que TS_OFFSET est le même pour tout horodatage tant que la valeur TS non adaptée ne revient pas à zéro ; voir le 4) ci-dessous.
2. Compression : Après l'initialisation, le compresseur ne compresse plus les valeurs de TS non adaptées. Il compresse plutôt les valeurs adaptées. La méthode de compression peut être le codage W-LSB ou le codage fondé sur le temporisateur.
3. Décompression : Lorsque il reçoit un TS_SCALED (compressé), le champ est d'abord décompressé, et l'horodatage RTP non adapté est alors calculé comme $TS = TS_SCALED * TS_STRIDE + TS_OFFSET$.
4. Décalage au retour à zéro : Si la valeur de TS_STRIDE n'est pas égale à une puissance de deux, le retour à zéro de l'horodatage non adapté de 32 bits va changer la valeur de TS_OFFSET. Quand cela arrive, le compresseur DEVRAIT réinitialiser TS_OFFSET en envoyant un horodatage non adapté, comme en 1 ci-dessus."

TEXTE INCORRECT ET INVALIDÉ DE LA RFC 3095 paragraphe 4.5.3 :

Le point 5 entier, c'est-à-dire tout le texte commençant par "5. Intervalle d'interprétation au moment du retour à zéro ...", jusque et y compris le bloc de texte qui commence par "Soit a le nombre de LSB nécessaire" et qui se termine par "... l'intervalle d'interprétation est b." est incorrect et est donc invalide.

4.5 Recalcul de TS_OFFSET

L'horodatage peut être envoyé non adapté si le changement de la valeur de TS ne correspond pas au TS_STRIDE établi, mais le TS_STRIDE pourrait quand même rester inchangé. Pour assurer une décompression correcte des paquets suivants, le décompresseur DOIT donc toujours recalculer TS_OFFSET (TS RTP modulo TS_STRIDE) quand un paquet avec une valeur d'horodatage non adaptée est reçu.

4.6 TS_STRIDE et le fanion Tsc dans l'extension 3

Le fanion Tsc dans Extension 3 indique si l'horodatage est adapté ou non. La valeur du fanion Tsc s'applique donc à tous les bits de l'horodatage, ainsi que si il n'y a pas de bits de TS dans l'extension elle-même. Quand TS est adapté, il est toujours adapté en utilisant contexte(TS_STRIDE). La légende de Extension 3 dans la RFC3095, paragraphe 5.7.5 déclare incorrectement que valeur(TS_STRIDE) est utilisé pour l'horodatage adapté.

Si TS_STRIDE est présent dans Extension 3, comme indiqué par le fanion Tss établi, l'en-tête compressé DEVRAIT porter des bits de TS non adapté ; c'est-à-dire, le fanion Tsc NE DEVRAIT PAS être établi quand Tss est établi car un TS non adapté est nécessaire avec TS_STRIDE pour recalculer le TS_OFFSET. Si TS_STRIDE est inclus dans un en-tête compressé avec un horodatage adapté, le décompresseur doit ignorer et éliminer le champ (TS_STRIDE).

TEXTE INCORRECT DE LA RFC 3095 paragraphe 5.7.5):

Tsc : Tsc = 0 indique que TS n'est pas adapté ;
 Tsc = 1 indique que TS est adapté selon le paragraphe 4.5.3, en utilisant valeur(TS_STRIDE).
 Contexte(Tsc) est toujours 1. Si l'adaptation n'est pas désirée, le compresseur établira TS_STRIDE = 1.

TEXTE CORRIGÉ :

Tsc : Tsc = 0 indique que TS n'est pas adapté ;
 Tsc = 1 indique que TS est adapté selon le paragraphe 4.5.3, en utilisant contexte(TS_STRIDE).
 Contexte(Tsc) est toujours 1. Si l'adaptation n'est pas désirée, le compresseur établira TS_STRIDE = 1.
 Si champ (Tsc) = 1, et si TSS = 1 (signifiant que TS_STRIDE est présent dans l'extension), champ (TS_STRIDE) DOIT être ignoré et éliminé."

Quand le compresseur re-établit une nouvelle valeur pour TS_STRIDE en utilisant Extension 3, il devrait envoyer des bits TS non adaptés avec TS_STRIDE.

4.7 Utilisation de la compression fondée sur un temporisateur

La compression fondée sur un temporisateur de l'horodatage RTP, comme décrite dans la RFC3095, paragraphe 4.5.4, peut être utilisée pour réduire le nombre de bits (octets) d'horodatage transmis nécessaire quand l'horodatage ne peut pas être déduit du numéro de séquence. La compression fondée sur un temporisateur est seulement utilisée pour la décompression d'en-têtes compressés qui contiennent un champ TS ; autrement, quand aucun bit d'horodatage n'est présent, l'horodatage est déduit linéairement du numéro de séquence (voir le paragraphe 4.2 de ce document).

Utiliser ou non la compression fondée sur le temporisateur est contrôlé par le champ de contrôle TIME_STRIDE, qui peut être réglé par un paquet IR, IR-DYN, ou un paquet compressé avec Extension 3. Avant que la compression fondée sur le temporisateur puisse être utilisée, le décompresseur doit informer le compresseur (canal par canal) sur sa résolution d'horloge en envoyant une option de retour CLOCK pour tout CID sur le canal. Le compresseur peut ensuite initier la compression fondée sur le temporisateur en envoyant (sur la base du contexte) un TIME_STRIDE non zéro au décompresseur. Quand le compresseur est sûr que le décompresseur a reçu la valeur de TIME_STRIDE, il peut passer à la compression fondée sur le temporisateur.

5. Compression de liste

5.1 Éléments de liste de CSRC dans la chaîne dynamique RTP

La RFC 3095, paragraphe 5.7.7.6 définit les parties statiques et dynamiques de l'en-tête RTP. Cette Section indique un champ "Liste générique de CSRC" dans la chaîne dynamique, qui a une longueur variable (voir la RFC3095, paragraphe 5.8.6). Ce champ est toujours d'au moins un octet, même si la liste est vide (à la différence de la liste de CSRC dans l'en-tête RTP non compressé, qui n'est pas présente quand le champ RTP CC est réglé à 0).

5.2 Occurrences multiples du champ CC

Les parties statiques et dynamiques de l'en-tête RTP sont définies dans la RFC3095, paragraphe 5.7.7.6. Dans la partie dynamique, un champ CC indique le nombre d'éléments CSRC présents dans la "Liste générique de CSRC". Un autre champ CC apparaît aussi dans la "Liste générique de CSRC" (RFC3095, paragraphe 5.8.6.1) parce que le type de codage 0 est toujours utilisé dans la chaîne dynamique. Les deux champs CC ont la même signification : la valeur du champ CC détermine le nombre d'éléments XI dans la liste de CSRC pour le type de codage 0, et elle n'est pas utilisée autrement. Donc ce qui suit s'applique :

AJOUT FORMEL À LA RFC 3095 :

"Le premier octet de la partie dynamique de l'en-tête RTP contient un champ CC, comme défini au paragraphe 5.7.7.6. Une seconde occurrence apparaît dans la "Liste générique de CSRC", qui est aussi dans la partie dynamique de l'en-tête RTP, où le type de codage 0 est utilisé conformément au format défini dans la RFC 3095, paragraphe 5.8.6.1.

Le compresseur DOIT régler les deux occurrences du champ CC à la même valeur.

Le décompresseur DOIT utiliser la valeur du champ CC provenant du type de codage 0 dans la liste générique de CRSC, et il DOIT donc ignorer la première occurrence du champ CC."

5.3 Gabarits binaires dans une compression de liste

L'insertion et/ou la suppression de schémas, décrite dans la RFC 3095, paragraphes 5.8.6.2 à 5.8.6.4, utilise des gabarits binaires pour indiquer les positions d'insertion ou de suppression au sein de la liste de références. La taille du gabarit binaire peut être de 7 bits ou de 15 bits.

Le compresseur PEUT utiliser un gabarit de 7 bits, même si la liste de référence a plus de sept éléments, pourvu que les changements à la liste soient seulement appliqués à des éléments dans les sept premiers de la liste de référence, laissant inchangés les éléments qui ont un index non couvert par le gabarit de 7 bits. Le décompresseur NE DOIT PAS modifier des éléments qui ont un index non couvert par le gabarit de 7 bits, quand un gabarit de 7 bits est reçu pour une liste de références qui contient plus de sept éléments.

5.4 En-têtes compressés avec compression de liste

La RFC 3095, paragraphe 5.8, déclare que les en-têtes qui peuvent faire partie de chaînes d'en-tête d'extension "incluent" les extensions AH [RFC4302], ESP NULL [RFC2410], d'encapsulation minimale (MINE) [RFC2004], GRE [RFC2784], [RFC2890], et IPv6 [RFC2460]. Cette liste des en-têtes qui peuvent être compressés est correcte, mais le mot "incluent" ne devrait pas être là, car seuls les types d'en-tête mentionnés peuvent en fait être traités. On devrait noter de plus que pour l'en-tête d'encapsulation minimale (MINE) il n'y a pas de discussion explicite de la façon de le compresser, car l'en-tête est envoyé soit non compressé, soit complètement compressé.

5.5 Compression de liste d'en-têtes ESP NULL

À cause du décalage des champs dans la partie de queue de l'en-tête ESP, un compresseur NE DOIT PAS compresser les paquets contenant plus d'un en-tête NUL ESP [RFC2410], sauf si le second en-tête le plus externe est traité comme un en-tête ESP régulier [RFC4303] et que les paquets sont compressés en utilisant le profil 0x0003.

5.6 Tableaux et indices de traduction pour les en-têtes d'extension IP

La RFC 3095, paragraphe 5.8.4 décrit comment les indices de listes sont associés aux éléments de listes et comment les listes de tableaux sont construites pour les en-têtes d'extension IP. Le texte déclare de façon incorrecte que un indice par type est utilisé, car le même type peut apparaître plusieurs fois avec un contenu différent dans une seule chaîne.

Dans la compression de liste d'en-têtes d'extension IP, un indice est associé à chaque en-tête d'extension individuel d'une chaîne d'en-têtes d'extension. Quand il y a plusieurs occurrences non identiques du même type d'extension (numéro de protocole) au sein d'une chaîne d'en-têtes, chacun DOIT avoir son propre indice.

Dans le cas où il y a plusieurs occurrences identiques du même type d'extension, le compresseur peut les associer au même indice. Quand la valeur d'un élément dont l'indice se produit plus d'une fois dans la liste est mise à jour, le compresseur DOIT envoyer la valeur pour chaque occurrence de cet indice dans la liste.

Quand le contenu des en-têtes d'extension change, une mise en œuvre peut choisir d'utiliser un indice différent ou de mettre à jour celui existant. Certaines extensions peuvent être compressées même quand certains champs changent, car ces changements peuvent être portés implicitement au décompresseur (par exemple, les numéros de séquence dans les en-têtes d'extension qui peuvent être déduits du SN RTP) ou explicitement (par exemple, au titre du champ En-tête d'extension IP dans Extension 3).

Quand il y a plus d'un en-tête IP, il y a plus d'une liste d'en-têtes d'extension, et un tableau de traduction est tenu pour chaque liste indépendamment des autres.

5.7 Liste de références

Une liste compressée en utilisant le type de codage 1 (insertion), le type 2 (suppression) ou le type 3 (suppression/insertion)

utilise un schéma de codage qui se fonde sur l'utilisation d'une liste de références dans le contexte (identifié comme `ref_id`).

Bien qu'il pourrait sembler être un bon choix d'envoyer une liste de type 1 quand `ref_id` est une liste vide, il n'y a rien à gagner à le faire par rapport à l'utilisation d'une liste de type 0. Envoyer une liste de type 2 quand `ref_id` est une liste vide conduirait à un échec, tandis que envoyer une liste de type 3 n'a pas de sens. Toutes ces solutions alternatives pourraient être vues comme possibles, selon la façon dont la compression de liste est spécifiée dans la RFC 3095.

Si ces alternatives étaient permises, un décompresseur serait obligé de tenir une fenêtre glissante des listes de `ref_id` en mode R, même pour le cas où aucun élément n'est envoyé dans la liste compressée, et ce n'est pas une exigence souhaitable. Utiliser le codage de liste de type 1, type 2, et type 3 n'est donc permis que pour les listes de références non vides.

AJOUT FORMEL À LA RFC 3095 :

"Sans considération du mode de fonctionnement, pour le codage de listes de type 1, type 2, et type 3, `ref_id` DOIT se référer à une liste non vide."

5.8. Compression de numéros de séquence AH et GRE

La RFC 3095, paragraphe 5.8.4.2 et paragraphe 5.8.4.4 décrit comment compresser l'en-tête d'authentification (AH, *Authentication Header*) [RFC4302] et l'encapsulation générique d'acheminement (GRE, *Generic Routing Encapsulation*) [RFC2784], [RFC2890]. Ces deux paragraphes présentent une possibilité d'omettre le numéro de séquence AH/GRE dans l'en-tête compressé, dans certaines circonstances. Cependant, les conditions spécifiques pour omettre le numéro de séquence AH/GRE, ainsi que les procédures concrètes de compression et décompression à appliquer, ne sont pas clairement définies pour garantir la robustesse et faciliter une mise en œuvre interopérable.

Les règles appropriées sont fournies dans le cas de ESP, c'est-à-dire :

"Numéro de séquence : non envoyé quand le décalage à partir du numéro de séquence de l'en-tête compressé est constant, quand le compresseur est sûr que le décompresseur a établi le décalage correct. Quand le décalage n'est pas constant, le numéro de séquence peut être compressé par l'envoi de LSB".

La même logique s'applique aux numéros de séquence AH/GRE.

TEXTE INCORRECT DE LA RFC 3095 paragraphe 5.8.4.2):

"Si le numéro de séquence dans le AH augmente linéairement lorsque le numéro de séquence RTP s'accroît, et si le compresseur est sûr que le décompresseur a obtenu le schéma, le numéro de séquence dans AH n'a pas besoin d'être envoyé. Le décompresseur applique l'extrapolation linéaire pour reconstruire le numéro de séquence dans l'AH."

TEXTE CORRIGÉ :

"Le numéro de séquence AH peut être omis de l'en-tête compressé quand le décalage à partir du numéro de séquence (SN) de l'en-tête compressé est constant, quand le compresseur est sûr que le décompresseur a établi le décalage correct."

TEXTE INCORRECT DE LA RFC 3095 paragraphe 5.8.4.4):

"Si le numéro de séquence dans l'en-tête GRE augmente de façon linéaire comme augmente le numéro de séquence RTP et si le compresseur est sûr que le décompresseur a reçu le schéma, le numéro de séquence dans GRE n'a pas besoin d'être envoyé. Le décompresseur applique une extrapolation linéaire pour reconstruire le numéro de séquence dans l'en-tête GRE."

TEXTE CORRIGÉ :

"Le numéro de séquence GRE peut être omis de l'en-tête compressé quand le décalage par rapport au numéro de séquence (SN) de l'en-tête compressé est constant, quand le compresseur est sûr que le décompresseur a établi le décalage correct."

6. Mise à jour de propriétés

6.1 Mises à jour implicites

Un paquet de mise à jour de contexte qui contient des informations de numéro de séquence compressées peut aussi porter des informations sur d'autres champs ; dans ce cas, ces champs sont mis à jour en accord avec le contenu du paquet. Le paquet de mise à jour met aussi implicitement à jour les champs induits (par exemple, Horodatage RTP) conformément au

mode en cours et à la fonction appropriée de transposition des champs mis à jour et induits.

Un paquet de mise à jour met donc à jour les valeurs de référence de tous les champs d'en-tête, explicitement ou implicitement, sauf pour le paquet UO-1-ID (voir le paragraphe 6.2 de ce document). En mode UO, tous les paquets sont des paquets de mise à jour, tandis qu'en mode R, tous les paquets avec un CRC sont des paquets de mise à jour.

Par exemple, un paquet UO-0 contient le numéro de séquence (SN) RTP compressé. Un tel paquet met aussi implicitement à jour l'horodatage RTP, l'identifiant IPv4, et les numéros de séquence des en-têtes d'extension IP.

6.2 Mise à jour des propriétés de UO-1*

La RFC 3095, paragraphe 5.7.3 déclare que les valeurs fournies dans les extensions portées dans un paquet UO-1-ID ne mettent pas à jour le contexte, sauf pour les champs SN, TS, ou IP-ID. Cependant, la RFC3095, paragraphe 5.8.1 déclare correctement que le tableau de traduction dans le contexte est mis à jour chaque fois qu'une paire (indice, élément) est reçue, ce qui est contredit par la déclaration du paragraphe 5.7.3 parce que le paquet UO-1-ID peut porter Extension 3 avec des éléments de paire (indice, élément) dans le champ "Liste de CSRC compressés". En plus de cette contradiction, le texte ne mentionne pas quoi faire des autres numéros de séquence déduits du SN, qui sont aussi à être implicitement mis à jour. Les propriétés de mises à jour de UO-1* comme déclarées par la RFC3095, paragraphe 5.7.3 sont donc incomplètes.

TEXTE INCOMPLET DE LA RFC 3095 paragraphe 5.7.3):

"Les valeurs fournies dans les extensions, sauf celles dans les autres champs SN, TS, ou IP-ID, ne mettent pas à jour le contexte."

TEXTE CORRIGÉ :

"Les paquets UO-1-ID mettent seulement à jour TS, SN, IP-ID, et les numéros de séquence des en-têtes d'extension IP. Les autres valeurs fournies dans les extensions ne mettent pas à jour le contexte."

Le décompresseur DOIT mettre à jour son tableau de traduction chaque fois qu'une paire (indice, élément) est reçue, conformément au paragraphe 5.8.1 de la RFC3095, et cette règle s'applique aussi aux paquets UO-1-ID."

6.3 Propriétés de mise à jour de contexte pour paquets IR

Les paquets IR ne suppriment pas tout le contexte, mais mettent à jour tous les champs portés dans l'en-tête IR. De même, un IR sans une chaîne dynamique met simplement à jour la partie statique du contexte, tandis que le reste du contexte est laissé inchangé.

Une conséquence de cela est que les champs qui ne sont pas mis à jour par le paquet IR, par exemple, les tableaux de traduction pour la compression de liste, NE DOIVENT PAS être invalidés par le décompresseur quand il suppose un dommage du contexte.

6.4 Champ de bourrage RTP (R-P) dans l'extension 3

La RFC 3095, paragraphe 5.7.5 définit les propriétés des fanions d'en-tête RTP et des champs dans Extension 3. Ils sont mis à jour quand le fanion rtp de l'Extension 3 est établi, c'est-à-dire, quand $rtp = 1$; autrement, ils ne sont pas mis à jour. Cependant, il n'est pas dit clairement comment Extension 3 met à jour le bit R dans le contexte.

TEXTE INCOMPLET DE LA RFC 3095 paragraphe 5.7.5):

"R-P : bit de bourrage RTP, valeur absolue (présumée zéro si absente)."

TEXTE CORRIGÉ :

"R-P : bit de bourrage RTP. Si $R-PT = 1$, R-P est la valeur absolue du bit de bourrage RTP et cette valeur met à jour le contexte (R-P). Si $R-PT = 0$, contexte (R-P) est mis à jour à zéro."

6.5 Bit eXtension (X) RTP dans la partie dynamique

La RFC 3095, paragraphe 5.7.7.6 définit les propriétés des fanions d'en-tête RTP et des champs dans la partie RTP de la chaîne dynamique de paquets IR et IR-DYN. Cependant, la façon dont le bit X bit est mis à jour dans le contexte n'est pas claire.

TEXTE INCOMPLET DE LA RFC 3095 paragraphe 5.7.7.6):

"X : Copie du bit X provenant de l'en-tête RTP (présupposé 0 si $RX = 0$)"

TEXTE CORRIGÉ :

"X : bit X provenant de l'en-tête RTP. Si $RX = 1$, X est le bit X bit provenant de l'en-tête RTP et cette valeur met à jour le contexte(X). Si $RX = 0$, contexte(X) est mis à jour à zéro."

7. Gestion de contexte et réutilisation de CID/contexte

7.1 Persistance de contextes de décompresseur

Au titre des paramètres de canal négociés, le compresseur et le décompresseur se sont accordés, par le paramètre `MAX_CID`, sur le plus haut numéro d'identification de contexte (CID) à utiliser. En s'accordant sur `MAX_CID`, le décompresseur accepte aussi de fournir des ressources de mémoire pour héberger au moins `MAX_CID+1` contextes, et un contexte établi avec un CID dans cet espace négocié DOIT être conservé par le décompresseur jusqu'à ce que le CID soit réutilisé, ou que le canal soit supprimé ou renégocié.

7.2 Réutilisation de CID/contexte

Au titre de la négociation du canal, le nombre maximal de contextes actifs acceptés est négocié entre le compresseur et le décompresseur par le paramètre `MAX_CID`. La valeur de `MAX_CID` peut différer significativement d'une application de liaison à l'autre, ainsi que la charge en terme de nombre de flux de paquets à compresser. La durée de vie d'un canal ROHC peut aussi varier, de presque permanent à une vie assez brève. Cependant, en général, on ne s'attend pas à ce que les ressources soient allouées pour plus de contextes qu'il peut être raisonnablement attendu qu'ils soient actifs concurremment sur la liaison. Par conséquent, les identifiants de contexte (CID) et la mémoire de contextes sont des ressources qui vont devoir être réutilisées par le compresseur au titre de ce qui peut être considéré comme du fonctionnement normal.

Comment les ressources de contexte sont réutilisées est non spécifié dans la [RFC3095] et les spécifications suivantes de ROHC. Le présent document n'a pas l'intention de changer cela, c'est-à-dire, la gestion de ressource de ROHC est toujours considérée comme un détail de mise en œuvre. Cependant, la réutilisation d'un CID et de son allocation de mémoire n'est pas toujours aussi simple que d'initier un contexte avec un CID précédemment non utilisé. Parce que certains profils peuvent fonctionner dans divers modes où les formats de paquets varient selon le mode courant, il faut faire attention de s'assurer que les données du vieux contexte vont être complètement écrasées en toute sécurité, éliminant le risque d'effets collatéraux non désirés résultant d'interactions entre les vieilles et les nouvelles données de contexte. Le présent document souligne donc certains aspects importants à considérer quand on met en œuvre la gestion de ressources dans les compresseurs et décompresseurs ROHC.

À haut niveau, la réutilisation de CID/contexte peut être de deux sortes, soit la réutilisation pour un nouveau contexte fondé sur le même profil que le vieux contexte, soit pour un nouveau contexte fondé sur un profil différent. Ces cas sont discutés séparément dans les deux paragraphes suivants.

7.2.1 Réutilisation d'un CID/contexte avec le même profil

Pour les profils multi modes, comme ceux définis dans la [RFC3095], les transitions de mode sont effectuées en utilisant une procédure de prise de contact initiée par le décompresseur, comme définie au paragraphe 5.6 de la RFC3095. Quand un CID/contexte est réutilisé pour un nouveau contexte fondé sur le même profil que le vieux contexte, le mode de fonctionnement en cours DEVRAIT être hérité du vieux au nouveau contexte. Précisément, le compresseur DEVRAIT continuer de fonctionner en utilisant le mode de fonctionnement du vieux contexte aussi avec le nouveau contexte. La raison en est qu'il n'y a pas de façon fiable pour le compresseur d'informer le décompresseur qu'une réutilisation de CID/contexte est en cours. Le décompresseur peut donc n'être pas supposé purger la mémoire de contexte pour le CID (voir le paragraphe 6.3) et il n'y a pas de moyen de déclencher un changement de mode sûr (qui exige que le décompresseur ait initié une procédure de prise de contact).

La règle de l'héritage de mode s'applique aussi quand le signal `CONTEXT_REINITIALIZATION` (RFC3095, paragraphe 6.3.1) est utilisé pour réinitialiser un contexte entier.

7.2.2 Réutilisation d'un CID/contexte avec un profil différent

Quand un CID est réutilisé pour un nouveau contexte fondé sur un profil différent de celui du vieux contexte, le compresseur et le décompresseur DOIVENT tous deux commencer à fonctionner avec ce contexte dans le mode initial du profil (si c'est un profil multi modes). Ceci s'applique aux nouveaux contextes initiés par IR et aux dégradations de profil avec IR-DYN (par exemple, la dégradation de profil 0x0001 en profil 0x0002 au paragraphe 5.11.1 de la RFC3095).

Les paquets de type 0 et de type 1 ont des formats différents dans le mode U/O et le mode R, et ces paquets de mode R n'ont pas de CRC. Quand on initie un nouveau contexte sur un CID réutilisé en mode R, il y a un risque que le décompresseur interprète mal les paquets compressés si les paquets IR initiateurs sont perdus.

Un CID pour un contexte fonctionnant actuellement en mode R DEVRAIT donc n'être pas réutilisé pour un nouveau contexte fondé sur un profil différent de celui du vieux contexte. Un compresseur qui fait autrement devrait minimiser le risque de mauvaise interprétation de R-0/R-1 en, par exemple, n'utilisant pas les paquets de types commençant par 00 ou 10 avant qu'il soit tout à fait sûr que le nouveau contexte a bien été initié au décompresseur.

8. Autres précisions au protocole

8.1 Signification de NBO

Dans la partie dynamique IPv4 (RFC3095, paragraphe 5.7.7.4) si le bit 'NBO' est établi, cela signifie que l'ordre des octets du réseau est utilisé.

8.2 IP-ID

Seon le paragraphe 5.7 de la RFC3095, IP-ID signifie la valeur compressée du champ 'Identification' de l'en-tête IPv4. Les paquets compressés contiennent cette valeur compressée (IP-ID) tandis que les paquets IR avec une chaîne dynamique et les paquets IR-DYN transmettent la valeur originale, non compressée de Identification. Le champ IP-ID représente toujours la valeur Identification de l'en-tête IPv4 le plus interne dont le fanion RND correspondant n'est pas 1.

Si RND ou RND2 est réglé à 1, le ou les IP-ID correspondants sont envoyés comme des valeurs de Identification de 16 bits non compressées à la fin de l'en-tête de base compressé, conformément à la description de IP-ID (voir le début du paragraphe 5.7 de la RFC3095). Quand il n'y a pas de IP-ID compressé, c'est-à-dire, pour IPv6 ou quand toutes les informations d'identification IP sont envoyées telles quelles (comme indiqué par RND/RND2 réglé à 1) le décompresseur ignore les bits IP-ID envoyés dans les en-têtes de base compressés.

Quand RND=RND2=0, IP-ID est compressé, c'est-à-dire, exprimé comme un décalage de SN et échange d'octets si NBO=0. C'est le cas aussi quand 16 bits de IP-ID sont envoyés dans Extension 3.

Quand RND=0 mais qu'aucun bit IP-ID n'est envoyé dans l'en-tête compressé, le décalage de SN pour IP-ID reste inchangé, ce qui signifie que Offset_m égale Offset_ref, comme décrit au paragraphe 4.5.5. Ceci est aussi exprimé d'une manière légèrement différente (avec la même signification) au paragraphe 5.7, où il est dit que "pente par défaut(décalage IP-ID) = 0", ce qui signifie que si aucun bit n'est envoyé pour IP-ID, sa pente de décalage de numéro de séquence prend la valeur de 0 par défaut.

8.3 Extension 3 dans les paquets UOR-2*

Certains fanions de l'en-tête IP dans l'extension (par exemple, NBO ou RND) peuvent changer l'interprétation des champs dans les paquets UOR-2*. Dans ce cas, quand un fanion change dans Extension 3, un décompresseur DOIT réanalyser le paquet UOR-2*.

8.4 Occurrences multiples du bit M

La partie en-tête RTP de Extension 3, comme définie au paragraphe 5.7.5 de la RFC3095, inclut un champ d'un bit pour le bit Marqueur RTP. Ce champ est aussi présent dans tous les formats d'en-tête de base compressé, sauf pour UO-1-ID; ce qui signifie qu'il peut y avoir deux occurrences du champ dans un seul en-tête compressé. Dans ce cas, les deux champs M doivent avoir la même valeur.

AJOUT FORMEL À LA RFC 3095 :

"Quand il y a deux occurrences du champ M dans un en-tête compressé (tous deux dans l'en-tête de base compressé et dans la partie RTP de Extension 3) le compresseur DOIT régler ces deux occurrences du champ M à la même valeur.

Au décompresseur, si les deux valeurs du champ M d'un tel paquet ne sont pas identiques, le paquet DOIT être éliminé."

8.5 Options SN multiples dans un paquet de rétroaction

La longueur du champ Numéro de séquence dans l'en-tête ESP original [RFC4303] est de 32 bits. Le format de l'option SN de retour (RFC3095, paragraphe 5.7.6.6) permet 8 bits additionnels de SN aux 12 bits de SN du format FEEDBACK-2 (RFC3095, paragraphe 5.7.6.1). Une seule option SN de retour n'est donc pas suffisante pour que le décompresseur renvoie tous les 32 bits du numéro de séquence ESP dans un paquet de retour, sauf si il utilise plusieurs options SN dans un paquet de retour.

Le paragraphe 5.7.6.1 de la RFC 3095 déclare qu'un paquet FEEDBACK-2 peut contenir un nombre variable d'options de retour, et que les options peuvent apparaître dans n'importe quel ordre.

Quand on traite plusieurs options SN dans un paquet de retour, le SN va être donné en enchaînant les champs.

8.6 Options de CRC multiples dans un paquet de rétroaction

Bien qu'il ne soit pas utile d'avoir plus d'une seule option CRC dans un paquet de retour, avoir plusieurs options CRC est quand même permis. Si plusieurs options CRC sont incluses, toutes les options CRC DOIVENT être identiques, car elles vont être calculées sur le même en-tête ; le compresseur DOIT autrement éliminer le paquet de retour.

8.7 Réponse à la perte de liaisons de rétroaction

Bien que ce ne soit ni désirable ni attendu, il peut arriver qu'une liaison utilisée pour porter des retours entre deux instances associées devienne indisponible. Si le compresseur peut être notifié d'un tel événement, le compresseur DEVRAIT redémarrer la compression pour chaque flux qui fonctionne en mode R. Quand il redémarre la compression, le compresseur DEVRAIT utiliser un CID différent pour chaque flux redémarré ; ceci est utile pour éviter la possibilité d'une mauvaise interprétation du type de l'en-tête compressé pour les identifiants de type de paquet qui sont communs aux deux modes U/O et R, quand le flux est redémarré en mode U (voir aussi le paragraphe 7.2).

Généralement, les liaisons de retour ne sont pas supposées disparaître une fois présentes, mais on devrait noter que ce pourrait être le cas pour certaines technologies de liaisons.

8.8 UOR-2 dans le profil 0x0002 (UDP) et le profil 0x0003 (ESP)

Un seul nouveau format est défini pour UOR-2 dans le profil 0x0002 et le profil 0x0003, qui remplace les trois formats (UOR-2, UOR-2-ID, UOR-2-TS) du profil 0x0001. Le même format UOR-2 est donc utilisé indépendamment de si il y a ou non des en-têtes IP avec un RND=1 correspondant. Cela s'applique aussi au profil IP [RFC3843] et au profil IP/UDP léger [RFC4019].

8.9 Bits de moindre poids du numéro de séquences dans les en-têtes d'extension IP

Dans la RFC 3095, paragraphe 5.8.5, les formats sont définis pour la compression des champs d'en-tête d'extension IP. Cela inclut des champs de numéro de séquence compressés, et ces champs contiennent le "LSB du numéro de séquence". Ces numéros de séquence ne sont pas "codés en LSB" comme, par exemple, le numéro de séquence RTP, mais ce sont les LSB des champs non compressés.

8.10 Attente d'accusé de réception UOR-2 en mode O

L'usage des ACK UOR-2 en mode O, comme discuté dans la RFC3095, paragraphe 5.4.1.1.2, est facultatif. Un décompresseur peut aussi envoyer des ACK pour d'autres besoins que d'accuser réception du UOR-2, sans avoir à continuer d'envoyer des ACK pour tous les UOR-2. De même, une mise en œuvre de compresseur peut ignorer les ACK de UOR-2 afin d'adapter des stratégies d'approche optimiste.

Il est donc **NON RECOMMANDÉ** d'utiliser le mécanisme de ACK facultatif en mode O, aussi bien dans les mises en œuvre de compresseur que de décompresseur.

Utiliser une attente incorrecte sur les ACK d'UOR-2 comme base du comportement d'un compresseur va dégrader de façon significative les performances de compression. C'est parce que les ACK UOR-2 peuvent être envoyés d'un décompresseur pour d'autres raisons que d'accuser réception du paquet UOR-2, par exemple, pour envoyer un retour comme la résolution d'horloge, ou initier une transition de mode. Si une mise en œuvre utilise l'algorithme facultatif d'accusé de réception décrit au paragraphe 5.4.1.1.2, elle doit s'assurer de régler les paramètres k_3 et n_3 à des valeurs bien supérieures à 1 pour s'assurer que les performances du compresseur ne sont pas dégradées à cause du problème décrit ci-dessus.

8.11 Réparations de contexte, TS_STRIDE et TIME_STRIDE

Le CRC de 7 bits utilisé pour vérifier le résultat de la décompression tente de couvrir l'en-tête original non compressé. La vérification du CRC exclut donc TS_STRIDE et TIME_STRIDE, car ces champs ne font pas partie de l'en-tête original non compressé.

Le type de paquet UOR-2 peut être utilisé pour mettre à jour la valeur de TS_STRIDE et/ou TIME_STRIDE, avec l'Extension 3. Cependant, ces champs ne sont pas utilisés pour la décompression du champ RTP TS pour ce type de paquet et leurs valeurs respectives ne sont donc pas vérifiées, implicitement ou explicitement.

Quand le compresseur reçoit un accusé de réception négatif, il ne peut donc pas déterminer si la défaillance peut être causée par un échec de mise à jour aux champs TS_STRIDE et/ou TIME_STRIDE, pour laquelle un en-tête précédent qui tentait de mettre à jour leur valeur a été précédemment acquitté.

AJOUT FORMEL À LA RFC 3095 :

"Quand le compresseur reçoit un NACK et utilise le type d'en-tête UOR-2 pour réparer le contexte du décompresseur, il DEVRAIT inclure les champs qui mettent à jour la valeur des deux TS_STRIDE et TIME_STRIDE dont il a mis à jour la valeur au moins une fois depuis l'établissement de ce contexte, c'est-à-dire, depuis que le CID a été associé à son profil courant.

Quand le compresseur reçoit un NACK statique, il DOIT inclure dans l'en-tête IR des champs pour les deux TS_STRIDE et le TIME_STRIDE dont il a mis à jour la valeur au moins une fois depuis l'établissement de ce contexte, c'est-à-dire, depuis que le CID a été associé à son profil courant."

9. Négociation de ROHC

La RFC 3095, paragraphe 4.1 déclare que la couche de liaison doit fournir des moyens pour négocier, par exemple, les paramètres de canal dont la liste figure au paragraphe 5.1.1 de la RFC3095. Un de ces paramètres est PROFILES, qui est un ensemble d'entiers non négatifs où chaque entier indique un profil supporté par le décompresseur.

Chaque profil est identifié par une valeur de 16 bits, où les 8 bits de moindre poids (LSB) indiquent le profil réel, et les 8 bits de poids fort (MSB) indiquent la variante de ce profil (voir la Section 8 de la RFC3095). Dans les en-têtes ROHC envoyés sur la liaison, le profil utilisé est identifié seulement par les 8 bits de LSB, ce qui signifie que le compresseur et le décompresseur doivent s'être mis d'accord sur la variante à utiliser pour chaque profil.

Le protocole de négociation doit donc être capable de communiquer au compresseur l'ensemble de profils supporté par le décompresseur. Quand plusieurs variantes du même profil sont disponibles, le protocole de négociation doit fournir le moyen pour que le décompresseur sache quelle variante va être utilisée par le compresseur. Cela signifie que l'ensemble de PROFILES après négociation NE DOIT PAS inclure plus d'une variante d'un profil.

10. Sous-option PROFILES dans ROHC sur PPP

L'union logique des sous options pour les négociations IPCP et IPV6CP, comme spécifié par ROHC sur PPP [RFC3241], ne peut pas être utilisée pour la sous option PROFILES, car toute l'union devrait alors être considérée au sein de chacune des deux négociations IPCP pour éviter d'obtenir un ensemble ambigu de profils. Une mise en œuvre de la RFC 3241 DOIT donc s'assurer que le même ensemble de profils est négocié pour IPv4 et IPv6 (IPCP/IPV6CP).

11. Codage IP-ID constant dans les profils IP seul et UPD léger

Dans le profil ROHC IP seul, au paragraphe 3.3 de la [RFC3843], un mécanisme pour coder une valeur constante d'identification dans IPv4 (IP-ID constant) est défini. Ce mécanisme est aussi utilisé par les profils ROHC UDP léger [RFC4019].

Le mécanisme "IP-ID constant" s'applique aux en-têtes IP internes et externes, quand ils sont présents, ce qui signifie qu'il va y avoir à la fois un SID et une valeur de contexte SID2.

12. Considérations sur la sécurité

Le présent document fournit un certain nombre de corrections et précisions à la [RFC3095], mais il ne fait aucun changement à l'égard des aspects de sécurité du protocole. Par conséquent, les considérations de sécurité de la [RFC3095] s'appliquent sans modification.

13. Remerciements

Les auteurs tiennent à remercier Vicknesan Ayadurai, Carsten Bormann, Mikael Degermark, Zhigang Liu, Abigail Surtees, Mark West, Tommy Lundemo, Alan Kennington, Remi Pelland, Lajos Zaccomer, Endre Szalai, Mark Kalmanczhelyi, et Arpad Szakacs de leurs contributions et commentaires. Merci aussi aux relecteurs désignés du document, Carl Knutsson et Biplab Sarkar, qui ont revu le document durant le dernier appel du groupe de travail.

14. Références

14.1 Références normatives

- [RFC1662] W. Simpson, éditeur, "[PPP en trames de style HDLC](#)", STD 51, juillet 1994. (*Remplace la RFC1549*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (*MàJ par RFC8174*)
- [RFC3095] C. Bormann et autres, "[Compression d'en-tête robuste](#) (ROHC) : cadre et quatre profils", juillet 2001. (*MàJ par RFC3759, RFC4815*) (*P.S.*)
- [RFC3241] C. Bormann, "[Compression d'en-tête robuste](#) (ROHC) sur PPP", avril 2002. (*MàJ par RFC4815*) (*P.S.*)
- [RFC3843] L-E. Jonsson, G. Pelletier, "[Compression d'en-tête robuste \(ROHC\)](#) : un profil de compression pour IP", juin 2004. (*MàJ par RFC4815*) (*P.S.*)
- [RFC4019] G. Pelletier, "[Compression d'en-tête robuste](#) (ROHC) : profils pour le protocole de datagramme d'utilisateur (UDP) léger", avril 2005. (*MàJ par RFC4815*) (*P.S.*)

14.2 Références pour information

- [RFC0768] J. Postel, "Protocole de [datagramme d'utilisateur](#) (UDP)", (STD 6), 28 août 1980.
- [RFC0791] J. Postel, éd., "Protocole Internet - Spécification du [protocole du programme Internet](#)", STD 5, septembre 1981.
- [RFC2004] C. Perkins, "[Encapsulation minimale au sein de IP](#)", octobre 1996. (*P.S.*)

- [RFC2410] R. Glenn, S. Kent, "L'algorithme de [chiffrement NULL](#) et son utilisation avec IPsec", novembre 1998. (P.S.)
- [RFC2460] S. Deering et R. Hinden, "Spécification du [protocole Internet, version 6](#) (IPv6)", décembre 1998. (MàJ par [5095](#), [6564](#) ; D.S ; Remplacée par [RFC8200](#), STD 86)
- [RFC2784] D. Farinacci, T. Li, S. Hanks, D. Meyer et P. Traina, "[Encapsulation d'acheminement générique](#) (GRE)", mars 2000.
- [RFC2890] G. Dommety, "[Extensions de clé et de numéro de séquence](#) à GRE", septembre 2000. (P.S.)
- [RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "[RTP : un protocole de transport pour les applications en temps réel](#)", STD 64, juillet 2003. (MàJ par [RFC7164](#), [RFC7160](#), [RFC8083](#), [RFC8108](#), [RFC8860](#))
- [RFC4302] S. Kent, "[En-tête d'authentification IP](#)", décembre 2005. (P.S.)
- [RFC4303] S. Kent, "[Encapsulation de charge utile](#) de sécurité dans IP (ESP)", décembre 2005. (Remplace [RFC2406](#)) (P.S.)
- [RFC4362] L-E. Jonsson et autres, "[Compression d'en-tête robuste](#) (ROHC) : Profil assisté de couche Liaison pour IP/UDP/RTP", janvier 2006. (Remplace [RFC3242](#)) (MàJ par [RFC4815](#)) (P.S.)

Appendice A. Exemple d'algorithme de CRC

```
#!/usr/bin/perl -w
use strict;
#=====
# ROHC CRC demo - Carsten Bormann cabo@tzi.org 2001-08-02
# Cette petite démonstration montre les quatre types de CRC utilisés dans la RFC 3095, la spécification pour la
# compression robuste d'en-tête. Taper les données en hexadécimal puis presser Control+D. #
#-----
#
# utility
#
sub dump_bytes($) {
    my $x = shift;
    my $i;
    for ($i = 0; $i < length($x); ) {
        printf("%02x ", ord(substr($x, $i, 1)));
        printf("\n") if (++$i % 16 == 0);
    }
    printf("\n") if ($i % 16 != 0);
}

#-----
#
# Algorithme de calcul de CRC.
#
sub do_crc($$$) {
    my $nbits = shift;
    my $poly = shift;
    my $string = shift;

    my $crc = ($nbits == 32 ? 0xffffffff : (1 << $nbits) - 1);
    for (my $i = 0; $i < length($string); ++$i) {
        my $byte = ord(substr($string, $i, 1));
        for (my $b = 0; $b < 8; $b++) {
            if (($crc & 1) ^ ($byte & 1)) {
                $crc >>= 1;
                $crc ^= $poly;
            }
        }
    }
}
```

```

    } else {
    $crc >>= 1;
    }
    $byte >>= 1;
  }
}
printf "%2d bits, ", $nbits;
printf "CRC: %02x\n", $crc;
}

#-----
#
# Séquence d'essai
#
$/ = undef;
$_ = <>; # lire jusqu'à EOF
my $string = ""; # extrait tout ce qui ressemble à de l'hexadécimal :
s/([0-9a-fA-F][0-9a-fA-F])/$string .= chr(hex($1)), ""/eg;
dump_bytes($string);

#-----
#
# CRC de segmentation à 32 bits
# Noter que le texte implique que ceci est complété comme pour PPP (cela diffère des CRC à 8, 7, et 3 bits)
#
#  $C(x) = x^0 + x^1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^{10} +$ 
#  $x^{11} + x^{12} + x^{16} + x^{22} + x^{23} + x^{26} + x^{32}$ 
#
do_crc(32, 0xedb88320, $string);

#-----
#
# 8-bit IR/IR-DYN CRC
#
#  $C(x) = x^0 + x^1 + x^2 + x^8$ 
#
do_crc(8, 0xe0, $string);

#-----
#
# 7-bit FO/SO CRC
#
#  $C(x) = x^0 + x^1 + x^2 + x^3 + x^6 + x^7$ 
#
do_crc(7, 0x79, $string);

#-----
#
# 3-bit FO/SO CRC
#
#  $C(x) = x^0 + x^1 + x^3$ 
#
do_crc(3, 0x6, $string);

```

Adresse des auteurs

Lars-Erik Jonsson
 Optand 737
 SE-831 92 Ostersund, Sweden

Kristofer Sandlund
 Ericsson AB
 Box 920

téléphone : +46 70 365 20 58
mél : lars-erik@lejonsson.com

Ghyslain Pelletier
Ericsson AB
Box 920
SE-971 28 Lulea, Sweden
téléphone : +46 8 404 29 43
mél : ghyslain.pelletier@ericsson.com

SE-971 28 Lulea, Sweden
téléphone : +46 8 404 41 58
mél : kristofer.sandlund@ericsson.com

Peter Kremer
Conformance and Software Test Laboratory
Ericsson Hungary
H-1300 Bp. 3., P.O. Box 107, HUNGARY
téléphone : +36 1 437 7033
mél : peter.kremer@ericsson.com

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2007)

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations 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, le IETF TRUST 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 encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

Remerciement

Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administratif de l'IETF (IASA).