

Groupe de travail Réseau  
**Request for Comments : 3611**  
 Catégorie : En cours de normalisation  
 Traduction Claude Brière de L'Isle

T. Friedman, Paris 6  
 R. Caceres, IBM Research  
 A. Clark, Telchemy  
 novembre 2003

## Rapports étendus du protocole de contrôle de RTP (RTCP XR)

### 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 (2003). Tous droits réservés.

### Résumé

Le présent document définit le type de paquet Rapport étendu (XR, *Extended Report*) pour le protocole de contrôle de RTP (RTPC, *RTP Control Protocol*) et définit comment l'utilisation des paquets XR peut être signalée par une application si elle emploie le protocole de description de session (SDP, *Session Description Protocol*). Les paquets XR sont composés de blocs de rapport, et sept types de blocs sont définis ici. L'objet du format de rapport étendu est de porter les informations qui complètent les six statistiques qui sont contenues dans les blocs de rapport utilisés par les paquets de rapport d'envoi (SR, *Sender Report*) et de rapport de réception (RR, *Receiver Report*) de RTCP. Certaines applications, telles que de surveillance des conséquences pour la diffusion groupée des caractéristiques du réseau (MINC, *multicast inference of network characteristics*) ou de voix sur IP (VoIP, *voice over IP*) exigent d'autres statistiques plus détaillées. En plus des types de bloc définis ici, des types de bloc supplémentaires pourront être définis à l'avenir qui adhèrent au cadre fourni par le présent document.

### Table des matières

1. Introduction.....	2
1.1 Applicabilité.....	3
1.2 Terminologie.....	4
2. Format de paquet de rapport d'extension.....	4
3. Cadre du bloc de rapport étendu.....	5
4. Blocs de rapport étendu.....	5
4.1 Bloc de rapport RLE perdu.....	5
4.2 Bloc de rapport RLE dupliqué.....	9
4.3 Bloc de rapport d'heure de réception de paquet.....	10
4.4 Bloc de rapport Heure de référence de receveur.....	12
4.5 Bloc de rapport DLRR.....	12
4.6 Bloc de rapport Statistiques sommaires.....	13
4.7 Bloc de rapport Métrique VoIP.....	15
5. Signalisation SDP.....	21
5.1 Attribut SDP.....	21
5.2 Usage en offre/réponse.....	23
5.3 Usage en dehors de offre/réponse.....	24
6. Considérations relatives à l'IANA.....	24
6.1 Type de paquet XR.....	24
6.2 Registre de type de bloc XR RTCP.....	25
6.3 Attribut SDP "rtcp-xr".....	25
7. Considérations pour la sécurité.....	26
A. Algorithmes.....	26
A.1 Interprétation du numéro de séquence.....	26
A.2 Exemple de calcul de perte de paquet de salve.....	27
Déclaration de droits de propriété intellectuelle.....	29
Références.....	29
Références normatives.....	29
Références pour information.....	30

Adresse des auteurs.....	30
Déclaration complète de droits de reproduction.....	31

## 1. Introduction

Le présent document définit le type de paquet Rapport étendu (XR, *Extended Report*) pour le protocole de contrôle de RTP (RTPC, *RTP Control Protocol*) [9], et définit comment l'utilisation des paquets XR peut être signalée par une application si elle emploie le protocole de description de session (SDP, *Session Description Protocol*) [4]. Les paquets XR portent des informations qui vont au delà de celles déjà contenues dans les blocs de rapport de réception des paquets de rapport d'envoi (SR) ou de rapport de réception (RR) de RTCP. Les informations sont utilisées par les profils RTP, et ne sont donc pas transportées de façon appropriée dans les extensions spécifiques de profil de SR ou RR. Les informations utilisées, par exemple pour la gestion du réseau, entrent dans cette catégorie.

La définition s'étale sur les trois sections qui suivent l'introduction. La Section 2 définit le paquet XR comme consistant en un en-tête de huit octets suivi par une série de composants appelés blocs de rapport. La Section 3 définit le format commun, ou cadre, consistant en un champ de type et un champ de longueur, exigé de tous les blocs de rapport. La Section 4 définit plusieurs types de bloc de rapport spécifiques. D'autres types de bloc pourront être définis dans de futurs documents lorsque le besoin s'en fera sentir.

Les types de bloc de rapport définis dans le présent document entrent dans trois catégories. La première catégorie consiste en rapports paquet par paquet sur les paquets RTP reçus ou perdus. Les rapports de la seconde catégorie portent des informations d'heure de référence entre les participants RTP. Dans la troisième catégorie, les rapports portent des mesures qui se rapportent à la réception des paquets, qui sont sommaires par nature mais qui sont plus détaillées, ou d'un type différent de celles qui sont convoyées dans les paquets RTCP existants.

Tout compris, sept formats de bloc de rapport sont définis dans ce document. Parmi eux, trois sont des types de bloc paquet par paquet :

- Bloc de rapport RLE perdu (paragraphe 4.1) : codage par plage des rapports concernant les pertes et réceptions de paquets RTP.
- Bloc de rapport RLE dupliqué (paragraphe 4.2) : codage par plage des rapports concernant les duplications de paquets RTP reçus.
- Bloc de rapport d'heure de réception de paquet (paragraphe 4.3) : Liste des horodatages de réception des paquets RTP.

Il y a deux types de bloc relatifs à l'heure de référence :

- Bloc de rapport de référence horaire de receveur (paragraphe 4.4) : Horodatage de l'horloge côté receveur. Conjointement au bloc de rapport DLRR mentionné ensuite, ils permettent aux non envoyeurs de calculer le délai d'aller-retour.
- Bloc de rapport DLRR (paragraphe 4.5) : Le délai depuis la réception du dernier bloc de rapport de référence horaire côté receveur. Un envoyeur de données RTP qui reçoit un bloc de rapport de référence horaire de receveur peut répondre par un bloc de rapport DLRR, tout a fait comme, dans le mécanisme déjà défini pour RTCP au paragraphe 6.3.1 de [9], un receveur de données RTP qui reçoit un horodatage NTP peut répondre en remplissant le champ DLSR d'un bloc de rapport de réception RTCP.

Finalement, le présent document définit deux types de bloc de métrique sommaire :

- Bloc de rapport de statistiques sommaires (paragraphe 4.6) : statistiques sur les numéros de séquence des paquets RTP, les pertes, les dupliqués, la gigue, et les valeurs de TTL ou de limites de bond.
- Bloc de rapport de métrique VoIP (paragraphe 4.7) : métriques pour la surveillance des appels en voix sur IP (VoIP).

Avant de procéder à la définition de paquet XR et bloc de rapport, le présent document donne une déclaration d'applicabilité (paragraphe 1.1) qui décrit le contexte dans lequel ces blocs de rapport peuvent être utilisés. Il définit aussi (paragraphe 1.2) l'utilisation normative des mots clés, tels que DOIT et DEVRAIT, comme ils sont employés dans le présent document.

À la suite des définitions des divers blocs de rapport, le document décrit comment les applications qui emploient SDP peuvent signaler leur utilisation (Section 5). Le document se conclut par une discussion (Section 6) sur les considérations de numérotation pour l'Autorité d'allocation des numéros de l'Internet (IANA), des considérations sur la sécurité (Section 7) et des appendices qui donnent des exemples de mise en œuvre des algorithmes exposés dans ce texte.

## 1.1 Applicabilité

Les paquets XR sont utiles sur plusieurs applications, et pour cette raison, ne sont pas définis comme extensions spécifiques de profil pour les rapports d'expéditeur ou destinataire RTCP (paragraphe 6.4.3 de [9]). Néanmoins, ils ne sont pas utiles dans tous les contextes. En particulier, le bloc de rapport de métrique VoIP (paragraphe 4.7) est spécifique des applications vocales, bien qu'il puisse être employé sur une large variété de telles applications.

Le bloc de rapport de métrique VoIP peut être appliqué à toute application vocale de un à un ou de un à plusieurs pour laquelle l'utilisation de RTP et RTCP est spécifiée. L'utilisation de métriques conversationnelles (paragraphe 4.7.5) y compris le facteur R (comme décrit par le modèle E défini dans [3]) et la note d'opinion moyenne pour la qualité de conversation (MOS-CQ, *mean opinion score for conversational quality*) dans les applications autres que des appels simples à deux parties n'est pas définie ; donc, ces métriques devraient être identifiées comme indisponibles dans les applications de conférence en diffusion groupée.

Les types de bloc de rapport paquet par paquet, RLE perdu (paragraphe 4.1) RLE dupliqué (paragraphe 4.2) et Heure de réception de paquet (paragraphe 4.3) ont été définis par référence à des applications de tomographie de réseau, telle que les conclusions pour la diffusion groupée de caractéristiques de réseau (MINC, *multicast inference of network characteristics*) [11]. MINC exige des traces détaillées de réception de paquet de la part des destinataires de session de diffusion groupée afin de déduire la structure globale de l'arborescence de distribution de diffusion groupée et de ses paramètres, tels que le taux de perte et de délais, qui s'appliquent aux chemins entre les points d'embranchement de cette arborescence.

Toute application multimédia de diffusion groupée en temps réel peut utiliser les types de bloc de rapport paquet par paquet. Une telle application pourrait employer un sous-système d'inférence MINC qui le fournirait avec les informations de topologie de l'arborescence de diffusion groupée. Une utilisation potentielle d'un tel sous-système serait pour l'identification de régions à forte perte dans l'arborescence de diffusion groupée et l'identification de participants à des sessions de diffusion groupée bien situés pour assurer la retransmission des paquets perdus.

Les rapports détaillés paquet par paquet n'ont pas nécessairement à consommer une bande passante disproportionnée par rapport aux autres paquets RTCP. Une application peut limiter la taille de ces blocs. Un mécanisme appelé "amincissement" est fourni pour ces blocs de rapport, et peut être utilisé pour assurer qu'ils respectent une limite de taille en restreignant le nombre de paquets faisant l'objet d'un rapport au sein de tout intervalle de numéro de séquence. Les raisons et l'utilisation de ce mécanisme sont décrites dans [13]. De plus, les applications peuvent ne pas exiger des blocs de rapport de la part de tous les destinataires afin de répondre à des questions importantes comme où, dans l'arborescence de diffusion groupée, se trouvent les chemins qui excèdent un seuil de taux de perte défini. Des décisions intelligentes concernant les destinataires qui ont envoyés ces blocs de rapport peuvent encore restreindre la portion de bande passante RTCP qu'ils consomment.

Les blocs de rapport paquet par paquet peuvent aussi être utilisés par des applications dédiées de surveillance du réseau. Pour de telles applications, il pourrait être approprié de permettre que plus de 5 % de la bande passante des données RTP soit utilisée pour les paquets RTCP, permettant ainsi des blocs de rapport proportionnellement plus grands et plus détaillés.

Rien dans les types de bloc paquet par paquet ne restreint leur utilisation aux applications de diffusion groupée. En particulier, ils pourraient être utilisés pour une tomographie de réseau similaire à MINC, mais utilisant à la place des paquets en envoi individuel rayés. De plus, si c'est estimé utile, ils pourraient être utilisés pour des applications limitées à deux participants.

Une utilisation pour laquelle les rapports paquet par paquet ne sont pas directement adaptés est celles des accusés de réceptions de paquet de données au titre du mécanisme de retransmission de paquet. La raison en est que la technique de comptabilisation des paquets suggérée pour ces blocs diffère de la comptabilité de paquet normalement employée par RTP. Afin de favoriser les applications de mesures, un effort est fait pour interpréter aussi peu que possible chez le destinataire des données, et de laisser autant que possible l'interprétation aux participants qui reçoivent les blocs de rapport. Donc, par exemple, un paquet avec un ID SSRC anormal ou un numéro de séquence anormal peut être exclu par la comptabilité RTP normale, mais pourrait être pris en compte pour les besoins de la surveillance du réseau.

Le bloc de rapport de statistiques sommaires (paragraphe 4.6) a aussi été défini en pensant à la surveillance du réseau. Ce type de bloc peut être utilisé également pour rapporter la réception de paquet en envoi individuel et en diffusion groupée.

Les types de bloc en relation avec l'heure de référence ont été conçus pour le contrôle d'encombrement de diffusion groupée orientée TCP fondée sur le destinataire [18]. En permettant aux destinataires de données de calculer leurs temps d'aller-retour avec les expéditeurs, ils aident les destinataires à estimer la bande passante vers l'aval qu'ils devraient demander. Noter que si chaque destinataire veut envoyer des blocs de rapport de temps de référence de destinataire (paragraphe 4.4) un expéditeur pourrait envoyer un nombre de blocs de rapport DLRR (paragraphe 4.5) égal au nombre de destinataires dont les paquets

RTCP sont arrivés chez l'envoyeur dans son intervalle de rapport. Lorsque augmente le nombre de participants à une session de diffusion groupée, une application devrait faire attention à quels participants envoient ces blocs, et à quelle fréquence.

Les paquets XR complètent les paquets RTCP existants, et peuvent être empilés avec les autres paquets RTCP pour former des paquets RTCP composés (voir la section 6 de [9]). L'introduction de paquets XR dans une session ne change en aucune façon les règles qui gouvernent le calcul de l'intervalle de rapport RTCP (voir le paragraphe 6.2 de [9]). Lorsque les paquets XR sont des paquets RTCP, ils comptent comme tels pour les calculs de bande passante. Il en résulte que l'addition d'informations de rapport étendu peut tendre à augmenter la taille moyenne de paquet RTCP, et donc l'intervalle moyen de rapport. Cette augmentation peut être limitée en limitant la taille des paquets XR.

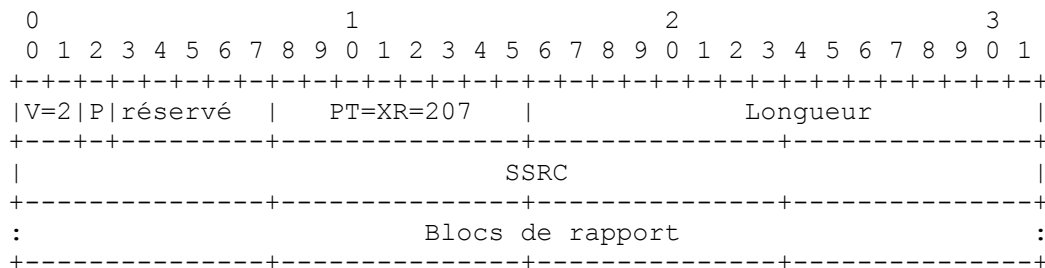
La signalisation SDP définie pour les paquets XR dans le présent document (Section 5) a été faite de telle sorte que trois scénarios soient possibles : une application de direct contrôlée par le protocole de flux directs en temps réel (RTSP, *Real Time Streaming Protocol*), une application multimédia de diffusion groupée de un à plusieurs comme un cours magistral avec des retours améliorés, et une session de conversation contrôlée par le protocole d'initialisation de session (SIP, *Session Initiation Protocol*) impliquant deux parties. Les applications qui emploient SDP sont libres d'utiliser de la signalisation SDP supplémentaire pour les cas non couverts ici. De plus, les applications ont toute liberté pour utiliser des mécanismes de signalisation autres que SDP.

## 1.2 Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans le BCP 14, RFC2119 [1] et indiquent les niveaux d'exigence pour la conformité à la présente spécification.

## 2. Format de paquet de rapport d'extension

Un paquet XR consiste en un en-tête de mots de 32 bits, suivi par un nombre, éventuellement zéro, de blocs de rapport étendu. Ce type de paquet est disposé de façon cohérente avec celle des autres paquets RTCP, en ce qui concerne la version essentielle, le type de paquet, et les informations de longueur. Les paquets XR sont donc rétro compatibles avec les mises en œuvre de receveur RTCP qui ne les reconnaissent pas, mais devraient être capables de les analyser en utilisant les informations de longueur. Un champ de bourrage et un champ SSRC sont aussi fournis dans les mêmes localisations qui apparaissent dans les autres paquets RTCP, par souci de simplicité. Le format est le suivant :



version (V) : 2 bits

Identifie la version de RTP. La présente spécification s'applique à RTP version deux.

Bourrage (P, *padding*) : 1 bit

Si le bit Bourrage est établi, ce paquet XR contient des octets supplémentaires de bourrage à la fin. La sémantique de ce champ est identique à celle du champ Bourrage dans les paquets SR, comme défini dans la spécification RTP.

réservé : 5 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Type de paquet (PT) : 8 bits

Contient la constante 207 pour identifier cela comme un paquet XR RTCP. Cette valeur est enregistrée auprès de l'IANA, comme décrit au paragraphe 6.1.

Longueur: 16 bits

Comme décrit pour le paquet Rapport d'envoyeur (SR) RTCP (voir le paragraphe 6.4.1 de la spécification RTP [9]). En bref, la longueur de ce paquet XR en mots de 32 bits moins un, incluant l'en-tête et tout bourrage.

SSRC : 32 bits

Identifiant de la source de synchronisation pour l'origine de ce paquet XR.

Blocs de rapport : longueur variable.

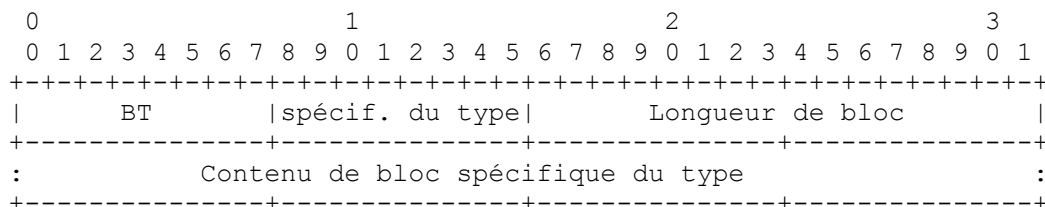
Zéro, un, ou plusieurs blocs de rapport étendu. En restant dans le cadre du bloc de rapport étendu défini ci-dessous, chaque bloc DOIT consister en un ou plusieurs mots de 32 bits.

### 3. Cadre du bloc de rapport étendu

Les blocs de rapport étendu sont mis en tas, l'un derrière l'autre, à la fin d'un paquet XR. La longueur d'un bloc individuel est un multiple de 4 octets. Le champ Longueur de l'en-tête XR décrit la longueur totale du paquet, incluant ces blocs de rapport étendu.

Chaque bloc a des champs type de bloc et longueur qui facilitent l'analyse. Une application receveuse peut démultiplexer les blocs sur la base de leur type, et peut utiliser les informations de longueur pour localiser chaque bloc successif, même en présence de types de bloc qu'elle ne reconnaît pas.

Un bloc de rapport étendu a le format suivant :



Type de bloc (BT) : 8 bits

Identifie le format de bloc. Sept types de bloc sont définis à la Section 4. Des types de bloc supplémentaires pourront être définis dans de futures spécifications. L'espace de noms de ce champ est géré par l'IANA, comme décrit à la Section 6.2.

Spécifique du type : 8 bits

L'utilisation de ces bits est déterminée par la définition du type de bloc.

Longueur de bloc : 16 bits

Longueur de ce bloc de rapport, incluant l'en-tête, en mots de 32 bits moins un. Si la définition du type de bloc le permet, zéro est une valeur acceptable, qui signifie un bloc qui consiste en seulement les champs BT, spécifique du type, et longueur de bloc, avec un champ Contenu de bloc spécifique du type nul.

Contenu de bloc spécifique du type : longueur variable.

L'utilisation de ce champ est définie par le type de bloc particulier, sous réserve de la contrainte qu'il DOIT être un multiple de 32 bits. Si la définition du type de bloc le permet, il PEUT être long de zéro bit.

## 4. Blocs de rapport étendu

Cette section définit sept blocs de rapport étendus : les types de bloc pour le rapport sur les pertes de paquets reçus et les paquets dupliqués, les heures de réception des paquets, les informations sur l'heure de référence du receveur, les délais inter rapport de receveur, les statistiques détaillées de réception, et les métriques de voix sur IP (VoIP). Une mise en œuvre DEVRAIT ignorer les blocs entrant avec des types non pertinents ou inconnus de lui. Les types de bloc supplémentaires DOIVENT être enregistrés auprès de l'Autorité d'allocation des numéros de l'Internet (IANA) [16], comme décrit au paragraphe 6.2.

### 4.1 Bloc de rapport RLE perdu

Ce type de bloc permet des rapports détaillés sur la réception des paquets individuels et les événements de pertes. De tels

rappports peuvent être utilisés, par exemple, pour les conclusions pour la diffusion groupée des caractéristiques du réseau (MINC) [11]. Avec les MINC, on peut découvrir la topologie de l'arborescence de diffusion groupée utilisée pour répartir les paquets RTP d'une source, et les taux de perte le long des liaisons au sein de cette arborescence, ou elles pourraient être utilisées pour fournir des données brutes à des applications de gestion d'un réseau.

Comme une trace booléenne de paquets RTP perdus et reçus est potentiellement lente, ce type de bloc permet que la trace soit compressée par un codage par plages. Pour réduire encore la taille de bloc, les rapports d'événements de pertes peuvent être systématiquement éliminés de la trace par un mécanisme appelé amincissement qui est décrit ci-dessous et qui est étudié dans [13].

Un participant qui génère un bloc de rapport RLE de perte devrait privilégier chaque fois que possible la précision dans les rapports d'événements observés plutôt que l'interprétation de ces événements. L'interprétation devrait être laissée à ceux qui observent les blocs de rapport. Suivre cette approche implique que la comptabilité de ces blocs de rapport de RLE de perte va différer de la comptabilité de la génération des paquets SR et RR décrits dans la spécification RTP [9] dans les deux domaines suivants : comptabilité par expéditeur et comptabilité par paquet.

Dans sa comptabilité par expéditeur, un participant à une session RTP NE DEVRAIT PAS faire de la réception d'un seuil minimum de nombre de paquets RTP une condition pour le rapport sur l'expéditeur de ces paquets. Cette technique comptable diffère de la technique décrite au paragraphe 6.2.1 et à l'Appendice A.1 de la spécification de RTP qui permet qu'un seuil détermine si un expéditeur est considéré comme valide.

Dans sa comptabilité par paquet, un participant à une session RTP DEVRAIT traiter tous les numéros de séquence comme valides. Cette technique de comptabilité diffère de la technique décrite à l'Appendice A.1 de la spécification RTP qui suggère de déclarer un numéro de séquence valide ou invalide sur la base de sa contiguïté avec les numéros de séquence des paquets reçus précédemment.

La validité de l'expéditeur et la validité du numéro de séquence sont des interprétations des données brutes. De telles interprétations sont justifiées dans l'intérêt, par exemple, d'exclure un vieux paquet égaré d'une session sans rapport pour l'empêcher d'avoir un effet sur le calcul de l'intervalle de transmission RTCP. La présence de paquets égarés pourrait, d'un autre côté, présenter de l'intérêt pour une application de surveillance du réseau.

Une interprétation comptable qui est toujours nécessaire est qu'un participant décide si les 16 bits du numéro de séquence sont revenus à zéro. Dans des circonstances ordinaires, ce n'est pas une tâche difficile. Par exemple, si le numéro de paquet 65 535 (le numéro de séquence le plus élevé possible) est suivi peu après par le paquet numéro 0, il est raisonnable de supposer qu'il y a eu un retour à zéro. Cependant, il est possible que le paquet soit antérieur (de 65 535 paquets plus tôt). Il est aussi possible que les numéros de séquence soient revenus à zéro plusieurs fois, soit en avant, soit en arrière. L'interprétation devient plus difficile lorsque il y a de gros intervalles entre les numéros de séquence, même en tenant compte du retour à zéro, et quand il y a de longs intervalles entre les paquets reçus.

La technique de comptabilité par paquet rendue obligatoire ici est qu'un participant garde trace du numéro de séquence du paquet reçu le plus récemment d'un expéditeur. Pour le prochain paquet qui arrive de cet expéditeur, le numéro de séquence DOIT être jugé ne pas tomber à plus de 32 768 en avant ou en arrière du plus récent, quel que soit le choix qui le place le plus près. Dans le cas où les deux choix sont également distants (ce qui n'est possible que lorsque la distance est 32 768) le choix DOIT être celui qui n'exige pas de retour à zéro. L'Appendice A.1 présente un algorithme qui met en œuvre cette technique.

Chaque bloc fait rapport d'une seule source de paquet de données RTP, identifiée par son SSRC. Le receveur qui fournit le rapport est identifié dans l'en-tête du paquet RTCP.

Le choix de commencer et finir les numéros de séquence de paquets RTP pour la trace est laissé à l'application. Ces valeurs sont rapportées dans le bloc. Le dernier numéro de séquence dans la trace PEUT différer du numéro de séquence rapporté dans tout rapport SR ou RR qui l'accompagne.

Noter qu'à cause du retour à zéro des numéros de séquence, le numéro de séquence qui termine PEUT être inférieur au numéro de séquence qui commence. Un bloc de rapport de RLE de perte NE DOIT PAS être utilisé pour rapporter sur une gamme de 65 534 ou plus dans l'espace des numéros de séquence, car il n'y a aucun moyen d'identifier plusieurs retours à zéro.

La trace décrite par un rapport RLE de perte consiste en une séquence de valeurs booléennes, une pour chaque numéro de séquence de la trace. Une valeur de un représente une réception de paquet, ce qui signifie que un ou plusieurs paquets qui ont ce numéro de séquence ont été reçus depuis le plus récent retour à zéro des numéros de séquence (ou depuis le commencement de la session RTP si aucun retour à zéro n'est jugé être intervenu). Une valeur de zéro représente une perte de paquet, ce qui signifie qu'il n'y a eu aucun paquet reçu pour ce numéro de séquence au moment du rapport. Si un paquet

avec un certain numéro de séquence est reçu après un rapport de perte pour ce numéro de séquence, un rapport RLE de perte PEUT rapporter un paquet reçu pour ce numéro de séquence.

Le codage lui-même consiste en une série d'unités de 16 bits appelés tronçons qui décrivent les séquences de paquets reçus ou perdus dans la trace. Chaque tronçon spécifie une plage ou un vecteur binaire, ou est un tronçon nul. Une plage décrit entre 1 et 16 383 événements qui sont tous les mêmes (tous des réceptions ou tous des pertes). Un vecteur binaire décrit 15 événements qui peuvent être un mélange de réceptions et de pertes. Un tronçon nul ne décrit aucun événement, et est utilisé pour arrondir le bloc à une limite de mot de 32 bits

La transposition d'une séquence de paquets perdus et reçus en une séquence de tronçons n'est pas nécessairement univoque. Par exemple, la trace suivante couvre 45 paquets, dont le 22<sup>ème</sup> et le 24<sup>ème</sup> ont été perdus et les autres reçus :

```
1111 1111 1111 1111 1111 1010 1111 1111 1111 1111 1111 1
```

Une façon de coder cela serait :

```
vecteur binaire 1111 1111 1111 111
vecteur binaire 1111 1101 0111 111
vecteur binaire 1111 1111 1111 111
tronçon nul
```

Une autre façon de coder cela serait :

```
plage de 21 reçus
vecteur binaire 0101 1111 1111 111
plage de 9 reçus
tronçon nul
```

Le choix du codage est laissé à l'application. Au titre de cette liberté de choix, les applications PEUVENT terminer une série de tronçons de plage de longueur et de vecteur binaire par un vecteur binaire ou tronçon qui va au delà de l'espace de numéro de séquence décrit par le bloc de rapport. Par exemple, si le 44<sup>ème</sup> paquet de la même séquence était perdu :

```
1111 1111 1111 1111 1111 1010 1111 1111 1111 1111 1110 1
```

Cela pourrait être codé comme :

```
plage de 21 reçus
vecteur binaire 0101 1111 1111 111
vecteur binaire 1111 1110 1000 000
tronçon nul
```

Dans cet exemple, les cinq derniers bits du second vecteur binaire décrivent une partie de l'espace de numéro de séquence qui s'étend au delà du dernier numéro de séquence de la trace. Ces bits ont été réglés à zéro.

Tous les bits dans un tronçon de vecteur binaire qui décrit une partie de l'espace de numéro de séquence s'étendant au delà du dernier numéro de séquence de la trace DOIVENT être réglés à zéro, et DOIVENT être ignorés par le receveur.

Un paquet nul DOIT apparaître à la fin d'un bloc de rapport RLE de perte si le nombre de tronçons de plages de longueur plus les tronçons de vecteur binaire est impair. Le tronçon nul NE DOIT PAS apparaître dans d'autres contextes.

Il faut faire attention lors de l'envoi de blocs de rapport RLE de perte parce que, même avec la compression fournie par le codage par plages, ils peuvent facilement consommer une largeur de bande passante hors de proportion avec celle des paquets RTCP normaux. Le type de bloc comporte un mécanisme, appelé amincissement, qui permet à une application de limiter les tailles de rapport.

Une valeur d'amincissement, T, choisit un sous-ensemble de paquets au sein de l'espace des numéros de séquence : ceux qui ont des numéros de séquence qui sont des multiples de  $2^T$ . Les rapports de perte et de réception de paquets ne s'appliquent qu'à ces paquets. T peut varier entre 0 et 15. Si T est zéro, chaque paquet dans l'espace de numéros de séquence fait alors l'objet d'un rapport. Si T est quinze, un seul paquet tous les 32 768 fait l'objet de rapport.

Supposons que la trace qu'on vient de décrire commence au numéro de séquence 13 821. Le dernier numéro de séquence dans la trace est 13 865. Si la trace devait être amincie avec une valeur d'amincissement de T = 2, les numéros de séquence suivants feraient alors l'objet d'un rapport : 13 824, 13 828, 13 832, 13 836, 13 840, 13 844, 13 848, 13 852, 13 856, 13 860, 13 864. La trace amincie serait comme suit :

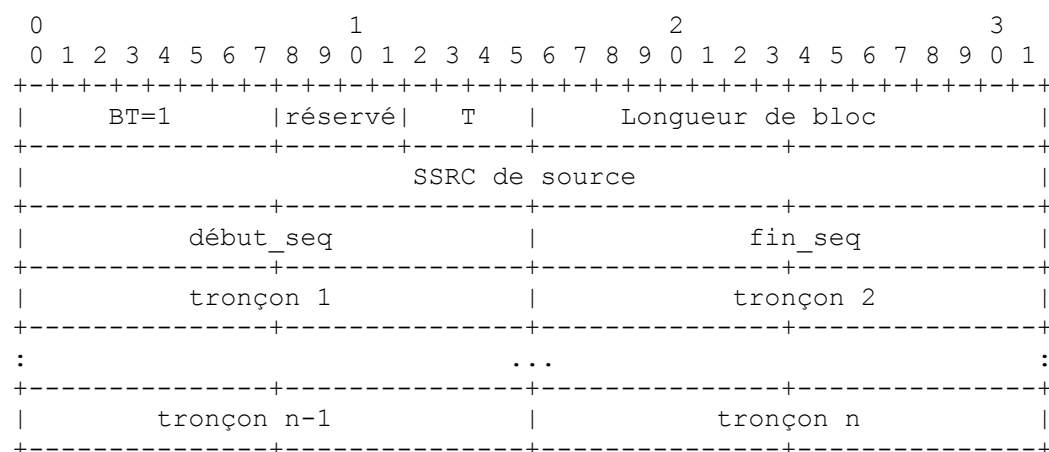
```
1 1 1 1 1 0 1 1 1 1 0
```

Ceci pourrait être codé comme suit :  
 vecteur binaire 1111 1011 1100 000  
 tronçon nul

Les quatre derniers bits dans le vecteur binaire, représentant les numéros de séquence 13 868, 13 872, 13 876, et 13 880, s'étendent au delà de la trace et sont donc réglés à zéro et sont ignorés par le receveur. Avec l'amincissement, la perte du 22<sup>ème</sup> paquet reste sans rapport parce que son numéro de séquence, 13 842, n'est pas un multiple de quatre. Les paquets reçus pour tous les numéros de séquence qui ne sont pas des multiples de quatre restent aussi sans rapport. Cependant, dans cet exemple, l'amincissement a permis que le bloc de rapport RLE de perte soit raccourci d'un mot de 32 bits.

Le choix de la valeur d'amincissement est laissé à l'application.

Le bloc de rapport RLE de perte a le format suivant :



Type de bloc (BT) : 8 bits

Un bloc de rapport RLE de perte est identifié par la constante 1.

réserve : 4 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Amincissement (T, *Thinning*) : 4 bits

Quantité d'amincissement effectuée sur l'espace de numéros de séquence. Seuls les paquets avec des numéros de séquence de  $0 \bmod 2^T$  font l'objet d'un rapport par ce bloc. Une valeur de 0 indique qu'il n'y a pas d'amincissement, et que tous les paquets font l'objet d'un rapport. L'amincissement maximum est un paquet tous les 32 768 (ce qui fait deux paquets dans chaque espace de séquence de 16 bits).

Longueur de bloc : 16 bits

Définie à la Section 3.

SSRC de source : 32 bits

C'est le SSRC de la source du paquet de données RTP dont il est fait rapport par ce bloc de rapport.

début\_seq : 16 bits

C'est le premier numéro de séquence dont ce bloc fait rapport.

fin\_seq : 16 bits

C'est le dernier numéro de séquence dont ce bloc fait rapport, plus un.

tronçon i : 16 bits

Il y a trois types de tronçon : plage de longueur, vecteur binaire, et terminaison nulle, définis dans les paragraphes qui suivent. Si le tronçon est tout de zéros, il est alors un tronçon de terminaison nulle. Autrement, le bit le plus à gauche du tronçon détermine son type : 0 pour une plage de longueur et 1 pour un vecteur binaire.



#### 4.1.1 Tronçon de plage de longueur

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+
|C|R|         Plage de longueur   |
+-----+-----+-----+-----+

```

Type de tronçon (C, *chunk type*) : 1 bit

Un zéro identifie ceci comme un tronçon de plage de longueur.

Type de plage (R, *run type*) : 1 bit

Zéro indique une plage de zéros. Un indique une plage de uns.

Plage de longueur : 14 bits

Une valeur entre 1 et 16 383. La valeur NE DOIT PAS être zéro pour un tronçon de plage de longueur (des zéros dans les deux champs type de plage et plage de longueur feraient du tronçon un tronçon de terminaison nul). Les plages de longueur de 15 ou moins PEUVENT être décrites par un tronçon de plage de longueur en dépit du fait qu'elles pourraient aussi être décrites au titre d'un tronçon vecteur binaire.

#### 4.1.2 Tronçon vecteur binaire

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+
|C|         vecteur binaire         |
+-----+-----+-----+-----+

```

Type de tronçon (C, *Chunk*) : 1 bit

Un un identifie ceci comme tronçon de vecteur binaire.

vecteur binaire : 15 bits

Le vecteur est lu de gauche à droite, dans l'ordre croissant de numéro de séquence (avec la tolérance appropriée pour le retour à zéro).

#### 4.1.3 Tronçon terminaison nulle

Ce tronçon est tout de zéros.

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+-----+-----+-----+
|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|
+-----+-----+-----+-----+

```

## 4.2 Bloc de rapport RLE dupliqué

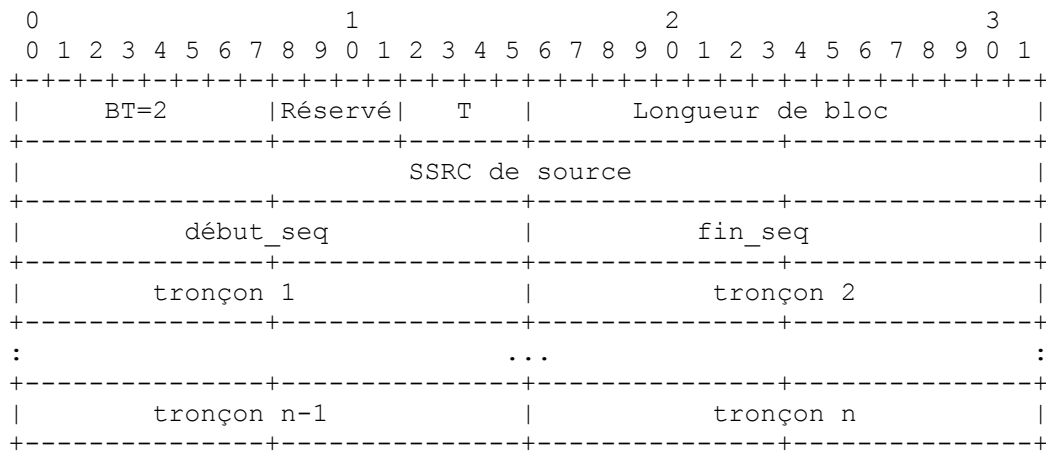
Ce type de bloc permet des rapports par numéro de séquence sur les dupliqués dans le flux de paquets RTP d'une source. De telles informations peuvent être utilisées pour des diagnostics du réseau, et fournissent une solution de remplacement pour les pertes de paquet comme base pour des conclusions sur la topologie de l'arborescence de diffusion groupée.

Le format de bloc de rapport de RLE dupliqué est identique au format de bloc de rapport RLE de perte. Seule l'interprétation est différente, en ce que les informations concernent les paquets dupliqués plutôt que les paquets perdus. La trace à coder dans ce cas consiste aussi en zéros et uns, mais ici, un zéro indique la présence de paquets dupliqués pour un certain numéro de séquence, tandis qu'un un indique qu'aucun dupliqué n'a été reçu.

L'existence d'un dupliqué pour un certain numéro de séquence est déterminé sur la période de rapport entière. Par exemple, si le paquet numéro 12 593 arrive, suivi par d'autres paquets avec d'autres numéros de séquence, l'arrivée plus tard dans la période de rapport d'un autre paquet numéroté 12 593 compte comme un dupliqué pour ce numéro de séquence. Le dupliqué n'a pas besoin de suivre immédiatement le premier paquet de ce numéro. Il faut veiller à ce qu'un rapport ne couvre pas une gamme de 65 534 ou plus dans l'espace des numéros de séquence.

Aucune distinction n'est faite entre l'existence d'un seul paquet dupliqué et celle de plusieurs paquets dupliqués pour un certain numéro de séquence. Noter aussi que comme il n'y a pas de duplication pour un paquet perdu, une perte est codée par un un dans un bloc de rapport de RLE dupliqué.

Le bloc de rapport de RLE dupliqué a le format suivant :



Type de bloc (BT) : 8 bits

Un bloc de rapport de RLE dupliqué est identifié par la constante 2.

Réservé : 4 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Amincissement (T, *Thinning*) : 4 bits

Comme défini au paragraphe 4.1.

Longueur de bloc : 16 bits

Défini à la Section 3.

SSRC de source : 32 bits

Comme défini au paragraphe 4.1.

début\_seq : 16 bits

Comme défini au paragraphe 4.1.

fin\_seq : 16 bits

Comme défini au paragraphe 4.1.

tronçon i : 16 bits

Comme défini au paragraphe 4.1.

### 4.3 Bloc de rapport d'heure de réception de paquet

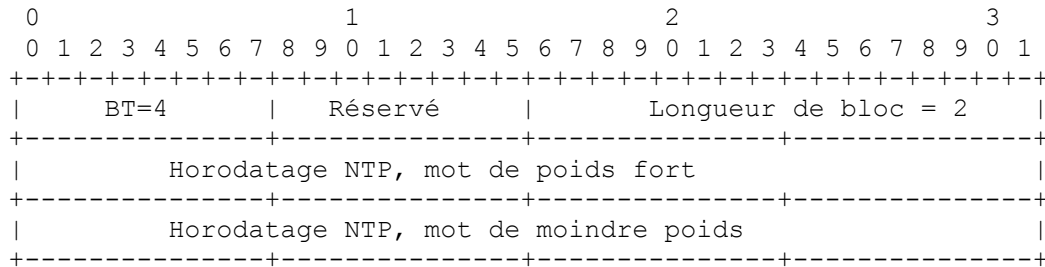
Ce type de bloc permet des rapports par numéro de séquence sur l'heure de réception des paquets pour un flux de paquets RTP d'une certaine source. De telles informations peuvent être utilisées pour des conclusions MINC sur la topologie de l'arborescence de diffusion groupée utilisée pour distribuer les paquets RTP de la source, et sur les délais le long des liaisons au sein de cette arborescence. Il peut aussi être utilisé pour mesurer les caractéristiques d'un chemin partiel et pour modéliser les distributions pour la gigue de paquet.

Les heures de réception de paquet sont exprimées dans les mêmes unités que dans les horodatages RTP des paquets de données. C'est fait de telle sorte que pour chaque paquet, on puisse établir en termes comparables l'heure d'envoi et l'heure de réception. Noter cependant que comme un envoyeur RTP initialise ordinairement son heure à une valeur choisie au hasard, on ne peut pas s'attendre à ce que les heures rapportées d'envoi et de réception diffèrent d'une quantité égale au délai unidirectionnel entre l'envoyeur et le receveur. Les heures rapportées peuvent néanmoins être utiles pour les besoins mentionnés précédemment.



#### 4.4 Bloc de rapport Heure de référence de receveur

Ce bloc étend les rapports d'horodatage de RTCP de façon que des non envoyeurs puissent aussi envoyer des horodatages. Il récapitule les champs d'horodatage NTP provenant du rapport d'envoyeur RTCP (voir le paragraphe 6.3.1 de [9]). Un non envoyeur peut estimer son délai d'aller-retour (RTT, *round trip time*) avec d'autres participants, comme proposé dans [18], en envoyant ce bloc de rapport et en recevant des blocs de rapport DLRR (voir au paragraphe suivant) en réponse.



Type de bloc (BT) : 8 bits

Un bloc de rapport Heure de référence de receveur est identifié par la constante 4.

Réservé : 8 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Longueur de bloc : 16 bits

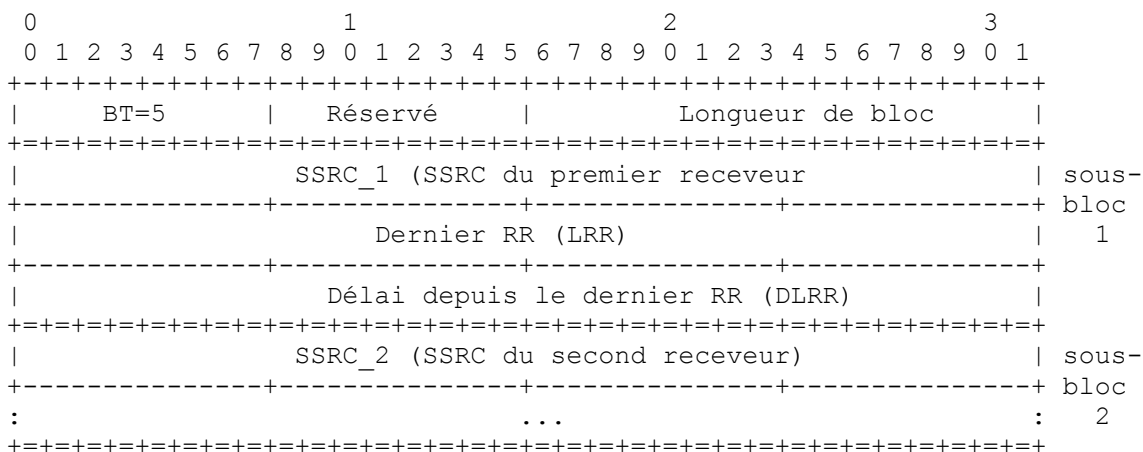
La constante 2, conformément à la définition de ce champ à la Section 3.

Horodatage NTP : 64 bits

Il indique l'heure d'horloge lorsque ce bloc a été envoyé, de telle sorte qu'il peut être utilisé en combinaison avec les horodatages retournés dans les blocs de rapport DLRR (voir le paragraphe suivant) provenant d'autres receveurs pour mesurer la propagation aller-retour avec ces receveurs. Les receveurs devraient s'attendre à ce que la précision de mesure de l'horodatage puisse être limitée à beaucoup moins que la résolution de l'horodatage NTP. L'incertitude de mesure de l'horodatage n'est pas indiquée car elle peut n'être pas connue. Un envoyeur de bloc de rapport qui peut garder trace du temps écoulé mais n'a pas de notion de l'heure d'horloge peut utiliser à la place le temps écoulé depuis le moment où il a rejoint la session. C'est supposé être moins de 68 ans, de sorte que le bit de poids fort sera zéro. Il est permis d'utiliser l'horloge d'échantillonnage pour estimer le temps écoulé à l'horloge. Un envoyeur de rapport qui n'a pas de notion de l'heure d'horloge ou du temps écoulé peut régler l'horodatage NTP à zéro.

#### 4.5 Bloc de rapport DLRR

Ce bloc étend le mécanisme de délai de RTCP depuis le dernier rapport d'envoyeur (DLSR, *Delay since the Last Sender Report*) (voir le paragraphe 6.3.1 de [9]) afin que les non envoyeurs puissent aussi calculer les temps d'aller-retour, comme proposé dans [18]. Il est appelé DLRR pour délai depuis le dernier rapport reçu, et peut être envoyé en réponse à un bloc de rapport d'horodatage de receveur (voir le paragraphe précédent) de la part d'un receveur pour permettre que ce receveur calcule son temps d'aller-retour avec le répondant. Le rapport consiste en un ou plusieurs sous-blocs de trois mots : un sous-bloc par rapport de receveur.



type de bloc (BT) : 8 bits

Un bloc de rapport DLRR est identifié par la constante 5.

Réservé : 8 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Longueur de bloc : 16 bits

Défini à la Section 3.

Horodatage du dernier RR (LRR) : 32 bits

Les 32 bits du milieu des 64 bits de l'horodatage NTP (comme expliqué au paragraphe précédent) reçus au titre d'un bloc de rapport d'heure de référence de receveur de la part du participant SSRC\_n. Si aucun de ces blocs n'a été reçu, le champ est réglé à zéro.

Délai depuis le dernier RR (DLRR) : 32 bits

C'est le délai, exprimé en unités de 1/65536 secondes, entre la réception du dernier bloc de rapport d'heure de référence de receveur de la part du participant SSRC\_n et l'envoi de ce bloc de rapport DLRR. Si un bloc de rapport d'heure de référence de receveur est encore à recevoir de la part de SSRC\_n, le champ DLRR est réglé à zéro (ou le DLRR est omis entièrement). Si SSRC\_r note le receveur qui produit le bloc de rapport DLRR, le participant SSRC\_n peut calculer le délai de propagation aller-retour au SSRC\_r en enregistrant le temps A lorsque ce bloc de rapport d'horodatage de receveur est reçu. Il calcule le temps total d'aller-retour A - LRR en utilisant le champ Dernier horodatage RR (LRR) puis soustrait ce champ pour laisser le délai de propagation aller-retour de A - LRR - DLRR. Ceci est illustré à la Figure 2 de [9].

#### 4.6 Bloc de rapport Statistiques sommaires

Ce bloc rapporte des statistiques au delà des informations portées dans le format de paquet RTCP standard, mais pas d'une granularité aussi fine que celles portées dans les blocs de rapport décrits précédemment. Les informations sont enregistrées sur les paquets perdus, les paquets dupliqués, les mesures de gigue, et les valeurs de TTL ou de limite de bonds. De telles informations peuvent être utiles pour la gestion du réseau.

Le contenu du bloc de rapport dépend d'une série de bits fanions portés dans la première partie de l'en-tête. Tous les paramètres n'ont pas besoin d'être rapportés dans chaque bloc. Les fanions indiquent lesquels sont rapportés ou non. Les champs correspondant aux paramètres non rapportés DOIVENT être présents, mais sont réglés à zéro. Le receveur DOIT ignorer tout bloc de rapport de statistiques sommaires avec une valeur différente de zéro dans tout champ dont le fanion indique qu'il n'y a pas de rapport.

Le bloc de rapport de statistiques sommaires a le format suivant :

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      BT=6      |L|D|J|ToH|rése.|      Longueur de bloc = 9      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SSRC de source                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      début_seq      |                                     fin_seq      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     paquets_perdus                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     paquets_dupliqués                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     gigue_mini                                       |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     gigue_maxi                                       |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     gigue_moyenne                                       |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     dev_gigue                                       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| min_ttl_ou_hl | max_ttl_ou_hl | moyn_ttl_ou_hl | dev_ttl_ou_hl |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

type de bloc (BT) : 8 bits

Un bloc de rapport de statistiques sommaires est identifié par la constante 6.

Fanion rapport de perte (L) : 1 bit

Bit réglé à 1 si le champ `paquets_perdus` contient un rapport, 0 autrement.

Fanion rapport de dupliqués (D) : 1 bit

Bit réglé à 1 si le champ `paquets_dupliqués` contient un rapport, 0 autrement.

Fanion gigue (J) : 1 bit

Bit réglé à 1 si les champs `gigue_mini`, `gigue_maxi`, `gigue_moyenne` et `dev_gigue` contiennent tous des rapports, 0 si aucun d'eux n'en contient.

Fanion TTL ou Limite de bond (ToH) : 2 bits

Ce champ est réglé à 0 si aucun des champs `min_ttl_ou_hl`, `max_ttl_ou_hl`, `moyen_ttl_ou_hl`, ou `dev_ttl_ou_hl` ne contient de rapport. Si le champ est non zéro, alors tous ces champs contiennent des rapports. La valeur 1 signifie qu'ils font rapport sur des valeurs de TTL IPv4. La valeur 2 signifie qu'ils font rapport sur des valeurs de limite de bonds IPv6. La valeur 3 est indéfinie et NE DOIT PAS être utilisée.

réservé : 3 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Longueur de bloc : 16 bits

La constante est 9, conformément à la définition de ce champ à la Section 3.

SSRC de source : 32 bits

Comme défini au paragraphe 4.1.

`début_seq` : 16 bits

Comme défini au paragraphe 4.1.

`fin_seq` : 16 bits

Comme défini au paragraphe 4.1.

`paquets_perdus` : 32 bits

Nombre de paquets perdus dans l'intervalle de numéro de séquence ci-dessus.

`paquets_dupliqués` : 32 bits

Nombre de paquets dupliqués dans l'intervalle de numéro de séquence ci-dessus.

`gigue_mini` : 32 bits

Temps de transit minimum relatif entre deux paquets dans l'intervalle de numéro de séquence ci-dessus. Toutes les valeurs de gigue sont mesurées comme la différence entre l'horodatage RTP d'un paquet et l'horloge du rapporteur au moment de l'arrivée, mesurés dans les mêmes unités.

`gigue_maxi` : 32 bits

Temps de transit maximum relatif entre deux paquets dans l'intervalle de numéro de séquence ci-dessus.

`gigue_moyenne` : 32 bits

Temps de transit moyen relatif entre chaque série de deux paquets dans l'intervalle de numéro de séquence ci-dessus, arrondi à la plus proche valeur exprimable comme horodatage RTP.

`dev_gigue` : 32 bits

Écart type du temps de transit relatif entre chaque série de deux paquets dans l'intervalle de numéro de séquence ci-dessus.

`min_ttl_ou_hl` : 8 bits

Valeur minimum du TTL ou de la limite de bonds des paquets de données dans la gamme de numéros de séquence.

`max_ttl_ou_hl` : 8 bits

Valeur maximum du TTL ou de la limite de bonds des paquets de données dans la gamme de numéros de séquence.

`moyen_ttl_ou_hl` : 8 bits

Valeur moyenne du TTL ou limite de bonds des paquets de données dans la gamme de numéros de séquence, arrondie à

l'entier le plus proche.

dev\_ttl\_ou\_hl : 8 bits

Écart type (*standard deviation*) des valeurs de TTL ou limite de bonds des paquets de données dans la gamme de numéros de séquence.

#### 4.7 Bloc de rapport Métrique VoIP

Le bloc de rapport de métrique VoIP fournit la métrique pour la surveillance des appels de voix sur IP (VoIP). Ces métriques incluent des métriques de perte et d'élimination de paquet, des métriques de délai, des métriques analogiques et des métriques de qualité de la voix. Le bloc fait rapport séparément sur les paquets perdus sur le canal IP, et ceux qui ont été reçus mais ensuite éliminés par la mémoire tampon de gigue du receveur. Il fait aussi rapport sur les effets combinés des pertes et éliminations, car toutes deux ont un effet égal sur la qualité de l'appel.

Pour retracer correctement la qualité d'un appel en voix sur IP, il est désirable de considérer le degré de sporadicité de la perte de paquet [14]. Suivant un modèle de Gilbert-Elliott [3], une période de temps, marquée par des paquets perdus et/ou éliminés avec un fort taux de pertes et/ou éliminations, est une "salve", et une période de temps entre deux salves est un "intervalle". Les salves correspondent aux périodes pendant lesquelles le taux de perte de paquet est assez élevé pour produire une dégradation remarquable de la qualité audio. Les intervalles correspondent aux périodes durant lesquelles seules des pertes de paquets isolés peuvent survenir, et en général elles peuvent être masquées par la technique de masquage de perte de paquet. Les rapports de délai incluent le délai de transit entre les points d'extrémité RTP et les délais de traitement du système VoIP terminal, dont tous les deux contribuent au délai perçu par l'utilisateur. Les métriques supplémentaires incluent les niveaux de signal, d'écho, de bruit, et de distorsion. Les métriques de qualité d'appel incluent le facteur R (comme décrit par le modèle E défini dans [6] et [3]) et les notes d'opinion moyenne (MOS, *mean opinion scores*).

Les mises en œuvre DOIVENT fournir des valeurs pour tous les champs définis ici. Pour certaines métriques, si la valeur est indéfinie ou inconnue, alors la valeur par défaut ou pour champ inconnu spécifiée DOIT être fournie.

Le bloc est codé comme sept mots de 32 bits :

```

      0                1                2                3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      BT=7      |  Réserve      |  Longueur de bloc = 8  |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SSRC de source                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Taux de perte | Tx d'éliminat.| Densité salve | Densité interv|
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Durée de salve   |   Durée d'intervalle   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Délai d'aller-retour   |   Délai système d'extrémité   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Niveau signal | Niveau bruit | RERL | Gmin |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Facteur R | ext. facteur R| MOS-LQ | MOS-CQ |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Config. RX | Réserve      |          JB nominal          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          JB maximum          |          JB abs max          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

type de bloc (BT) : 8 bits

Un bloc de rapport de métrique VoIP est identifié par la constante 7.

Réserve : 8 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

Longueur de bloc : 16 bits

La constante 8, conformément à la définition de ce champ à la Section 3.





La salve consiste en les douze paquets indiqués ci-dessus, commençant à un paquet éliminé et se terminant à un paquet perdu. Le premier intervalle commence au début de la session et le second intervalle se termine au moment du rapport.

Si l'espacement de paquet est de 10 ms et si la valeur Gmin est celle recommandée de 16, la durée de la salve est 120 ms, la densité de salve de 0,33, la durée d'intervalle de 290 ms = 520 ms, et la densité d'intervalle de 0,04.

Il en résulterait les valeurs rapportées comme suit (voir les descriptions de champ pour la sémantique et les détails de la façon dont ils sont calculés) :

taux de perte	12, qui correspond à 5 %
taux d'élimination	12, qui correspond à 5 %
densité de salve	84, qui correspond à 33 %
densité d'intervalle	10, qui correspond à 4 %
durée de salve	120, valeur en millisecondes
durée d'intervalle	520, valeur en millisecondes

densité de salve : 8 bits

C'est la fraction des paquets de données RTP qui, au sein des périodes de salve depuis le début de la réception, ont été perdus ou éliminés. Cette valeur est exprimée comme un nombre à virgule fixe avec la virgule binaire à gauche du champ. Elle est calculée en divisant le nombre total de paquets perdus ou éliminés (en excluant les paquets dupliqués éliminés) au sein des périodes de salve par le nombre total de paquets attendu au sein des périodes de salve, en multipliant le résultat de la division par 256, en limitant la valeur maximum à 255 (pour éviter le débordement) et en prenant la partie entière. Ce champ DOIT être rempli et DOIT être réglé à zéro si aucun paquet n'a été reçu.

densité d'intervalle : 8 bits

C'est la fraction des paquets de données RTP qui, au sein des intervalles entre les salves depuis le début de la réception, ont été perdus ou éliminés. Cette valeur est exprimée comme un nombre à virgule fixe avec la virgule binaire à gauche du champ. Elle est calculée en divisant le nombre total de paquets perdus ou éliminés (en excluant les paquets dupliqués éliminés) au sein des périodes d'intervalle par le nombre total de paquets attendus au sein des périodes d'intervalle, en multipliant le résultat de la division par 256, en limitant la valeur maximum à 255 (pour éviter le débordement) et en prenant la partie entière. Ce champ DOIT être rempli et DOIT être réglé à zéro si aucun paquet n'a été reçu.

durée de salve : 16 bits

Durée moyenne, exprimée en millisecondes, des périodes de salve qui sont survenues depuis le début de la réception. La durée de chaque période est calculée sur la base des paquets qui marquent le début et la fin de cette période. Elle est égale à l'horodatage du paquet de fin, plus la durée du paquet de fin, moins l'horodatage du paquet de début. Si les valeurs réelles ne sont pas disponibles, les valeurs estimées DOIVENT être utilisées. Si il n'y a pas eu de période de salve, la valeur de la durée de salve DOIT être zéro.

durée d'intervalle : 16 bits

C'est la durée moyenne, exprimée en millisecondes, des périodes d'intervalle qui sont survenues depuis le début de la réception. La durée de chaque période est calculée sur la base du paquet qui marque la fin de la précédente salve et le paquet qui marque le début de la salve suivante. Elle est égale à l'horodatage du paquet de salve suivant, moins l'horodatage du paquet de salve précédent, plus la durée du paquet de salve précédent. Si les valeurs réelles ne sont pas disponibles, les valeurs estimées DOIVENT être utilisées. Dans le cas d'un intervalle qui survient au début de la réception, la somme de l'horodatage du précédent paquet de salve et la durée du précédent paquet de salve sont remplacées par l'heure de début de réception. Dans le cas d'un intervalle qui survient à la fin de la réception, l'horodatage du paquet de salve suivant est remplacé par l'heure de fin de réception. Si il n'y a pas eu de période d'intervalle, la valeur de durée d'intervalle DOIT être zéro.

### 4.7.3 Métriques de délai

Pour les besoins des définitions qui suivent, l'interface RTP est l'interface entre l'instance RTP et l'application vocale (c'est-à-dire, la FEC, le désentrelaçage, le démultiplexage, la mémoire tampon de gigue). Par exemple, le retard dû au multiplexage de charge utile RTP serait considéré comme faisant partie du retard de l'application vocale ou du système d'extrémité, tandis que le retard dû au multiplexage des trames RTP au sein d'une trame UDP serait considéré comme faisant partie du délai RTP rapporté. Cette distinction est cohérente avec l'utilisation de RTCP pour les mesures de délais.

délai d'aller-retour : 16 bits

C'est le plus récent délai d'aller-retour calculé entre les interfaces RTP, exprimé en millisecondes. Cette valeur PEUT être mesurée en utilisant RTCP, la méthode DLRR définie au paragraphe 4.5 du présent document, lorsque il est nécessaire de

convertir les unités de mesure de valeurs d'horodatage NTP en millisecondes, ou selon d'autres approches. Si RTCP est utilisé, alors la valeur de délai rapportée est l'heure de réception du plus récent paquet RTCP provenant du SSRC de source, moins l'heure du LSR (dernier SR) rapportée dans son SR (rapport d'envoi), moins le DLSR (délai depuis le dernier SR) rapporté dans son SR. Une valeur de LSR non zéro est exigée afin de calculer le délai d'aller-retour. Une valeur de 0 est permise ; cependant, ce champ DOIT être rempli aussitôt qu'une estimation de délai est disponible.

délai de système terminal : 16 bits

C'est la plus récente estimation du délai du système terminal, exprimée en millisecondes. Le délai de système terminal est défini comme la somme du délai total d'accumulation et de codage d'échantillons associé à la direction d'envoi et du délai de mémoire tampon de gigue, de décodage, et de mémoire tampon d'exécution associé à la direction de réception. Ce délai PEUT être estimé ou mesuré. Cette valeur DEVRAIT être fournie dans tous les rapports de métrique VoIP. Si une mise en œuvre n'est pas capable de fournir les données, la valeur 0 DOIT être utilisée.

Noter que le délai de segment VoIP symétrique unidirectionnel peut être calculé à partir des délais d'aller-retour et de système terminal comme suit ; si le délai d'aller-retour est noté RTD, et si les délais de système terminal associés à deux points d'extrémité sont ESD(A) et ESD(B), alors :

$$\text{délai de chemin vocal unidirectionnel symétrique} = ( \text{RTD} + \text{ESD(A)} + \text{ESD(B)} ) / 2$$

#### 4.7.4 Métriques en rapport avec le signal

Les métriques suivantes sont destinées à fournir des informations en temps réel en relation avec les éléments non paquets du système de voix sur IP pour aider à l'identification des problèmes qui affectent la qualité de l'appel. Les valeurs identifiées ci-dessous doivent être déterminées pour le signal audio reçu. Les informations requises pour remplir ces champs peuvent n'être pas disponibles dans tous les systèmes, bien qu'il soit fortement recommandé que ces données DEVRAIENT être fournies pour prendre en charge les diagnostics de problèmes.

Niveau du signal : 8 bits

Le niveau relatif du signal vocal est défini comme le rapport du niveau du signal à une référence de 0 dBm0 [10], exprimée en décibels par un entier signé en forme de complément à deux. Ceci n'est mesuré que pour les paquets qui contiennent de l'énergie vocale. L'intention de cette métrique n'est pas de fournir une mesure précise du niveau de signal mais de donner une indication en temps réel que le niveau de signal peut être excessivement élevé ou faible.

$$\text{niveau de signal} = 10 \text{ Log}_{10} ( \text{écart quadratique moyen (rms) de puissance d'émission de parole (mW)} )$$

Une valeur de 127 indique que ce paramètre est indisponible. Les valeurs normales devraient généralement être dans la gamme -15 à -20 dBm.

niveau de bruit : 8 bits

Le niveau de bruit est défini comme le rapport du niveau de bruit de fond de période de silence à une référence de 0 dBm0, exprimé en décibels par un entier signé en forme de complément à deux.

$$\text{niveau de bruit} = 10 \text{ Log}_{10} ( \text{écart quadratique moyen de la puissance de silence (mW)} )$$

Une valeur de 127 indique que ce paramètre est indisponible.

affaiblissement d'adaptation pour l'écho résiduel (RERL, *residual echo return loss*) : 8 bits

La valeur d'affaiblissement d'adaptation pour l'écho résiduel peut être mesurée directement par l'annuleur d'écho du système VoIP d'extrémité ou peut être estimée en ajoutant les valeurs d'affaiblissement d'adaptation pour l'écho (ERL) et d'amélioration d'affaiblissement d'adaptation pour l'écho (ERLE) rapportées par l'annuleur d'écho.

$$\text{RERL(dB)} = \text{ERL (dB)} + \text{ERLE (dB)}$$

Dans le cas d'une passerelle VoIP, la source d'écho est normalement l'écho de ligne qui survient aux points de conversion de 2 à 4 fils dans le réseau. Cela peut être dans la gamme 8-12 dB. Un annuleur d'écho de ligne peut donner un ERLE de 30 dB ou plus et donc réduire cela à 40-50 dB. Dans le cas d'un téléphone IP, cela peut être le couplage acoustique entre le combiné écouteur et microphone ou l'écho résiduel acoustique provenant du fonctionnement du microphone, et peut plus correctement être appelé affaiblissement de couplage terminal (TCL, *terminal coupling loss*). Un combiné normal résulterait en un affaiblissement d'écho de 40-50 dB dû au retour acoustique.

Exemples :

- Passerelle IP connectée à un réseau à commutation de circuits avec une boucle à deux fils. Sans annulation d'écho, un convertisseur 2-4 fils normal a un ERL de 12 dB :  $RERL = ERL + ERLE = 12 + 0 = 12$  dB.
- Passerelle IP connectée à un réseau à commutation de circuits avec une boucle à deux fils. Avec annuleur d'écho cela améliore l'écho de 30 dB :  $RERL = ERL + ERLE = 12 + 30 = 42$  dB.
- Téléphone IP avec combiné conventionnel. Le couplage acoustique de l'écouteur au microphone du combiné (affaiblissement de couplage terminal) est normalement de 40 dB :  $RERL = TCL = 40$  dB.

Si on note A l'extrémité locale du chemin VoIP et B l'extrémité distante, et si l'équivalent pour la sonie à l'émission (SLR, *sender loudness rating*) et l'équivalent pour la sonie à la réception (RLR, *receiver loudness rating*) sont connus pour A (valeurs par défaut respectivement de 8 dB et 2 dB) alors le niveau d'affaiblissement d'écho à l'extrémité A (équivalent pour la sonie d'écho pour la personne qui parle ou TELR (*talker echo loudness rating*)) est donné par :

$$\begin{aligned} \text{TELR}(A) &= \text{SRL}(A) + \text{ERL}(B) + \text{ERLE}(B) + \text{RLR}(A) \\ \text{TELR}(B) &= \text{SRL}(B) + \text{ERL}(A) + \text{ERLE}(A) + \text{RLR}(B) \end{aligned}$$

Donc, pour incorporer l'écho dans une estimation de la qualité de la voix à l'extrémité A d'une connexion VoIP, il est désirable d'envoyer la valeur  $\text{ERL} + \text{ERLE}$  de B à A en utilisant un format comme XR de RTCP.

Les informations qui se rapportent à l'écho peuvent n'être pas disponibles dans tous les systèmes d'extrémité VoIP. Comme l'écho a un effet significatif sur la qualité de conversation, il est recommandé que des valeurs estimées pour l'affaiblissement d'adaptation d'écho et pour l'affaiblissement de couplage terminal soient fournies (si des estimations cohérentes peuvent être raisonnablement déterminées).

Des valeurs typiques pour les systèmes d'extrémité sont données ci-dessous à titre d'indication :

- Téléphone IP avec combiné : normalement 45 dB.
- Téléphone logiciel sur PC ou casque et microphone : extrêmement variable, envisager de rapporter "indéfini" (127).
- Passerelle IP avec annuleur d'écho de ligne : ERL et ERLE sont normalement disponibles.
- Passerelle IP sans annuleur d'écho de ligne : source fréquente de problèmes d'écho, envisager de rapporter une valeur faible (12 dB) ou "indéfinie" (127).

Gmin Voir le paragraphe 4.7.6 "Paramètres de configuration" ci-dessous.

#### 4.7.5 Métriques de qualité d'appel ou de transmission

Les métriques suivantes sont des mesures directes de la qualité d'appel ou de transmission, et incorporent les effets du type de codec, de perte de paquet, d'élimination, de sporadicité, de délai, etc. Ces métriques peuvent ne pas être disponibles dans tous les systèmes, cependant, elles DEVRAIENT être fournies afin de prendre en charge les diagnostics de problèmes.

facteur R : 8 bits

Le facteur R est une métrique de qualité de la voix qui décrit le segment de l'appel qui est porté sur cette session RTP. Il est exprimé par un entier dans la gamme 0 à 100, une valeur de 94 correspondant à "qualité de type circuit interurbain" et des valeurs de 50 ou moins étant considérées comme inutilisables. Cette métrique est définie comme incluant les effets du délai, en cohérence avec la Recommandation UIT-T G.107 [6] et la spécification technique ETSI TS 101 329-5 [3].

Une valeur de 127 indique que ce paramètre est indisponible. Les valeurs autres que 127 et la gamme de validité définie ci-dessus NE DOIVENT PAS être envoyées et DOIVENT être ignorées par le système receveur.

facteur externe R : 8 bits

Le facteur externe R est une métrique de qualité de la voix qui décrit le segment de l'appel qui est porté sur un segment de réseau externe au segment RTP, par exemple un réseau cellulaire. Ses valeurs sont interprétées de la même manière que pour le facteur R RTP. Cette métrique est définie comme incluant les effets du délai, en cohérence avec la Recommandation UIT-T G.107 [6] et la spécification technique ETSI TS 101 329-5 [3], et se rapporte au chemin sortant de la voix à partir de la terminaison de voix sur IP pour laquelle ce bloc de métrique s'applique.

Une valeur de 127 indique que ce paramètre est indisponible. Les valeurs autres que 127 et la gamme de validité définie ci-dessus NE DOIVENT PAS être envoyées et DOIVENT être ignorées par le système receveur.

Noter qu'un facteur R global peut être estimé à partir du facteur R du segment RTP et du facteur R externe, comme suit :

$$R \text{ total} = \text{facteur R RTP} + \text{facteur R externe} - 94$$

MOS-LQ : 8 bits

La note d'opinion moyenne estimée de qualité pour la personne qui écoute (MOS-LQ) est une métrique de qualité de la voix sur une échelle de 1 à 5, dans laquelle 5 représente excellent et 1 représente inacceptable. Cette métrique est définie comme n'incluant pas les effets du délai et peut être comparée aux notes de MOS obtenues des essais de qualité pour la personne qui écoute (ACR) tests. Il est exprimé par un entier dans la gamme de 10 à 50, correspondant à MOS x 10. Par exemple, une valeur de 35 correspondrait à une note de MOS estimée de 3,5.

Une valeur de 127 indique que ce paramètre est indisponible. Les valeurs autres que 127 et la gamme valide définie ci-dessus NE DOIVENT PAS être envoyées et DOIVENT être ignorées par le système receveur.

MOS-CQ : 8 bits

La note d'opinion moyenne estimée pour la qualité de conversation (MOS-CQ) est définie comme incluant les effets du délai et autres effets qui affecteraient la qualité de conversation. La métrique peut être calculée en convertissant un facteur R déterminé conformément à la Recommandation UIT-T G.107 [6] ou la spécification technique ETSI TS 101 329-5 [3] en une MOS estimée en utilisant l'équation spécifiée dans G.107. Elle est exprimée par un entier dans la gamme de 10 à 50, correspondant à MOS x 10, comme pour MOS-LQ.

Une valeur de 127 indique que ce paramètre est indisponible. Les valeurs autres que 127 et la gamme valide définie ci-dessus NE DOIVENT PAS être envoyées et DOIVENT être ignorées par le système receveur.

#### 4.7.6 Paramètres de configuration

Gmin : 8 bits

C'est le seuil d'intervalle. Ce champ contient la valeur utilisée pour ce bloc de rapport pour déterminer si un intervalle existe. La valeur recommandée de 16 correspond à une période de salve qui a une densité minimum de 6,25 % de paquets perdus ou éliminés, qui peut causer une dégradation perceptible de la qualité d'appel; durant les périodes d'intervalle, si une perte de paquet ou une élimination survient, chaque paquet perdu ou éliminé serait précédé et suivi par une séquence d'au moins 16 paquets reçus non éliminés. Noter que les paquets perdus ou éliminés qui surviennent au sein de Gmin paquets d'un rapport qui est généré peuvent être reclassés au titre d'une salve ou intervalle dans des rapports ultérieurs. La spécification technique ETSI TS 101 329-5 [3] définit un algorithme de calcul efficace pour mesurer la densité de salves et d'intervalles en utilisant une approche fondée sur les événements de perte/élimination de paquets. Cet algorithme est reproduit à l'Appendice A.2 du présent document. Gmin NE DOIT PAS être zéro, DOIT être fourni, et DOIT rester constant sur les blocs de rapport de métrique VoIP pour la durée la session RTP.

octet de configuration du receveur (RX config) : 8 bits

Cet octet comporte les champs suivants :

```

0 1 2 3 4 5 6 7
+-----+
| PLC | JBA | taux JB |
+-----+
```

masquage de perte de paquet (PLC, *packet loss concealment*) : 2 bits

Standard (11) / amélioré (10) / désactivé (01) / non spécifié (00). Lorsque PLC = 11, un algorithme simple de répétition ou d'interpolation est alors utilisé pour remplir le paquet manquant ; cette approche est normalement capable de masquer les paquets perdus isolés à des taux faibles de perte de paquets. Lorsque PLC = 10, un algorithme amélioré d'interpolation est alors utilisé ; les algorithmes de ce type sont capables de masquer efficacement de forts taux de perte. Lorsque PLC = 01, du silence est alors inséré à la place des paquets perdus. Lorsque PLC = 00, aucune information n'est alors disponible concernant l'utilisation de PLC ; cependant, pour certains codecs, cela peut être déduit.

mémoire tampon de gigue adaptable (JBA, *jitter buffer adaptive*) : 2 bits

Adaptable (11) / non adaptable (10) / réservé (01) / inconnu (00). Lorsque la mémoire tampon de gigue est adaptable, sa taille est alors ajustée de façon dynamique pour traiter les divers niveaux de gigue. Lorsqu'elle est non adaptable, la taille de mémoire tampon de gigue est maintenue à un niveau fixe. Lorsque les modes adaptable ou non adaptable sont spécifiés, les paramètres de taille de mémoire tampon de gigue ci-dessous DOIVENT être spécifiés.

taux de mémoire tampon de gigue (taux JB) : 4 bits

J = taux d'ajustement (0-15). Cela représente le taux d'ajustement spécifique de mise en œuvre d'une mémoire tampon de gigue en mode adaptable. Ce paramètre est défini en termes de temps approximatif qu'il faut pour s'ajuster pleinement à un changement d'étape d'une gigue de crête à crête de 30 ms à 100 ms telle que :

$$\text{temps d'ajustement} = 2 * J * \text{taille de trame (ms)}$$

Ce paramètre est destiné seulement à fournir une indication du degré "d'agressivité" d'une mémoire tampon de gigue adaptable et peut être estimé. Une valeur de 0 indique que le temps d'ajustement est inconnu pour cette mise en œuvre.

réservé : 8 bits

Ce champ est réservé pour une future définition. En l'absence d'une telle définition, les bits de ce champ DOIVENT être réglés à zéro et DOIVENT être ignorés par le receveur.

#### 4.7.7 Paramètres de mémoire tampon de gigue

Les valeurs rapportées dans ces champs DEVRAIENT être les plus récemment obtenues au moment du rapport.

délai nominal de mémoire tampon de gigue (JB nominal) : 16 bits

C'est le délai nominal actuel de mémoire tampon de gigue en millisecondes, qui correspond au délai nominal de mémoire tampon de gigue pour les paquets qui arrivent exactement à l'heure. Ce paramètre DOIT être fourni pour les mises en œuvre de mémoire tampon de gigue aussi bien fixes qu'adaptables.

délai maximum de mémoire tampon de gigue (JB maximum) : 16 bits

C'est le délai maximum actuel de mémoire tapon de gigue en millisecondes qui correspond au paquet arrivé le plus tôt qui ne serait pas éliminé. Dans une mise en œuvre de simple file d'attente, cela peut correspondre à la taille nominale. Dans les mises en œuvre de mémoire tampon de gigue adaptables, cette valeur peut varier de façon dynamique jusqu'à JB abs max (voir ci-dessous). Ce paramètre DOIT être fourni pour les mises en œuvre de mémoire tampon de gigue aussi bien fixes qu'adaptables.

délai maximum absolu de mémoire tampon de gigue (JB abs max) : 16 bits

C'est le délai maximum absolu en millisecondes que la mémoire tampon adaptable peut atteindre dans les pires conditions. Si cette valeur excède 65 535 millisecondes, alors ce champ DEVRA porter la valeur 65 535. Ce paramètre DOIT être fourni pour les mises en œuvre de mémoire tampon de gigue adaptables et sa valeur DOIT être réglée à JB maximum pour les mises en œuvre de mémoire tampon de gigue fixes.

## 5. Signalisation SDP

La présente section définit la signalisation du protocole de description de session (SDP, *Session Description Protocol*) [4] pour les blocs XR qui peuvent être employés par les applications qui utilisent SDP. Cette signalisation est définie pour être utilisée soit par les applications qui mettent en œuvre le modèle d'offre/réponse SDP [8] ou par les applications qui utilisent SDP pour décrire les configurations de support et de transport en connexion avec des protocoles comme le protocole d'annonce de session (SAP, *Session Announcement Protocol*) [15] ou le protocole de flux directs en temps réel (RTSP, *Real Time Streaming Protocol*) [17]. Il existe d'autres méthodes potentielles de signalisation qui ne sont pas définies ici.

Les blocs XR PEUVENT être utilisés sans signalisation préalable. Cela est cohérent avec les règles qui gouvernent les autres types de paquet RTCP, comme décrites dans [9]. Un exemple dans lequel la signalisation ne serait pas utilisée est une application qui exige toujours l'utilisation d'un ou plusieurs blocs XR. Cependant, pour les applications qui sont configurées à l'initialisation de la session, l'utilisation d'un type de signalisation est recommandée.

Noter que, bien que l'utilisation de la signalisation SDP pour les blocs XR puisse être facultative, si elle est utilisée, elle DOIT être utilisée comme défini ici. Si la signalisation SDP est utilisée dans un environnement où les blocs XR ne sont mis en œuvre que par une fraction des participants, ceux qui ne mettent pas en œuvre les blocs XR vont ignorer l'attribut SDP.

### 5.1 Attribut SDP

Ce paragraphe définit un nouvel attribut SDP "rtcp-xr" qui peut être utilisé pour signaler aux participants à une session support qu'ils devraient utiliser les blocs XR spécifiés. Cet attribut peut facilement être étendu à l'avenir avec de nouveaux paramètres pour couvrir tous nouveaux blocs de rapport.

L'attribut SDP Blocs XR RTCP XR est défini ci-dessous en forme Backus-Naur augmenté (ABNF) [2]. C'est à la fois un attribut de niveau session et support. Lorsque spécifié au niveau session, il s'applique à tous les blocs de niveau support dans la session. Toute spécification de niveau support DOIT remplacer une spécification de niveau session, si il en est une présente, pour ce bloc de support.

```

rtcp-xr-attr = "a=" "rtcp-xr" ":" [xr-format *(SP xr-format)] CRLF
xr-format = pkt-loss-rle / pkt-dup-rle / pkt-rcpt-times / rcvr-rtt / stat-summary / voip-metrics / format-ext
pkt-loss-rle = "pkt-loss-rle" ["=" max-size]
pkt-dup-rle = "pkt-dup-rle" ["=" max-size]
pkt-rcpt-times = "pkt-rcpt-times" ["=" max-size]
rcvr-rtt = "rcvr-rtt" "=" rcvr-rtt-mode [":" max-size]
rcvr-rtt-mode = "all" / "sender"
stat-summary = "stat-summary" ["=" stat-flag *("," stat-flag)]
stat-flag = "loss" / "dup" / "jitt" / "TTL" / "HL"
voip-metrics = "voip-metrics"
max-size = 1*DIGIT ; taille maximum de bloc en octets
DIGIT = %x30-39
format-ext = non-ws-string
non-ws-string = 1*(%x21-FF)
CRLF = %d13.10

```

L'attribut "rtcp-xr" contient zéro, un, ou plusieurs paramètres qui se rapportent aux blocs XR. Chaque paramètre signale des fonctionnalités pour un bloc XR, ou un groupe de blocs XR. L'attribut est extensible de sorte que les paramètres peuvent être définis pour tout futur bloc XR (et un paramètre devrait être défini pour chaque futur bloc).

Chaque paramètre "rtcp-xr" appartient à une de deux catégories. La première catégorie, les paramètres unilatéraux, est pour les blocs de rapport qui rapportent simplement sur le flux RTP et les métriques qui s'y rapportent. La seconde catégorie, les paramètres collaboratifs, est pour les blocs XR qui impliquent des actions de plus d'une partie afin de porter leurs fonctions.

La mesure du temps d'aller-retour (RTT, *round trip time*) est un exemple de fonction collaborative. Un receveur de paquet de données RTP envoie un bloc de rapport d'heure de référence (paragraphe 4.4). Un participant qui reçoit ce bloc envoie un bloc de rapport DLRR (paragraphe 4.5) en réponse, permettant au receveur de calculer son RTT avec ce participant. Comme cet exemple l'illustre, la fonction collaborative peut être mise en œuvre par deux ou plus blocs XR différents. La fonction collaborative de plusieurs blocs XR peut être gouvernée par un seul paramètre "rtcp-xr".

Pour la catégorie unilatérale, le présent document définit les paramètres suivants. Les noms des paramètres et leur format XR correspondant sont les suivants :

Nom de paramètre	Bloc XR (type de bloc et nom)
pkt-loss-rle	1 bloc de rapport RLE de perte
pkt-dup-rle	2 bloc de rapport RLE dupliqué
pkt-rcpt-times	3 bloc de rapport Heure de réception de paquet
stat-summary	6 bloc de rapport Statistiques sommaires
voip-metrics	7 bloc de rapport Métriques VoIP

Les paramètres "pkt-loss-rle", "pkt-dup-rle", et "pkt-rcpt-times" PEUVENT spécifier une valeur d'entier. Cette valeur indique la plus grande taille que le bloc de rapport entier DEVRAIT avoir en octets. Cela devra être vu comme une indication que l'amincissement devra être appliqué si nécessaire pour satisfaire à la taille cible.

Le paramètre "stat-summary" contient une liste qui indique quels champs DEVRAIENT être inclus dans les blocs de rapport Statistiques sommaires qui sont envoyés. Les éléments de la liste sont séparés par des virgules, et elle contient un ou plusieurs indicateurs de champs. Le caractère espace (0x20) NE DEVRA PAS être présent au sein de la liste. Les indicateurs de champ représentent les fanions définis au paragraphe 4.6. Les indicateurs de champ et leurs fanions respectifs sont les suivants :

Indicateur	Fanion
perte	fanion rapport de perte (L)
dup	fanion rapport de dupliqué (D)
gigue	fanion gigue (J)
TTL	fanion TTL ou Limite de bond (ToH)
HL	fanion TTL ou Limite de bond (ToH)

Pour "perte", "dup", et "gigue", la présence de l'indicateur indique que le fanion correspondant devrait être réglé à 1 dans les blocs de rapport de Statistiques sommaires qui sont envoyés. La présence de "TTL" indique que le fanion correspondant devrait être réglé à 1. La présence de "HL" indique que les fanions correspondants devraient être réglés à 2. Les indicateurs "TTL" et "HL" NE DOIVENT PAS être signalés ensemble.

Les blocs de la catégorie collaborative sont classés comme blocs initiateurs ou blocs de réponse. La signalisation DEVRAIT indiquer quels participants sont obligés de répondre au bloc initiateur. Une partie qui souhaite recevoir des blocs de réponse de ces participants peut déclencher cela en envoyant un bloc initiateur.

La catégorie collaborative consiste actuellement en une seule fonctionnalité, à savoir le mécanisme de mesure du temps d'aller-retour pour les receveurs de données RTP. La fonction collective du bloc de rapport Heure de référence de receveur et du bloc de rapport DLRR est représentée par le paramètre "rcvr- rtt". Ce paramètre prend comme arguments une valeur de mode et, facultativement, une taille maximum pour le bloc de rapport DLRR. La valeur de mode "tout" indique qu'aussi bien les envoyeurs que les receveurs de données RTP PEUVENT envoyer des blocs DLRR, tandis que la valeur de mode "envoyeur" indique que seuls les envoyeurs RTP actifs PEUVENT envoyer des blocs DLRR, c'est-à-dire que les envoyeurs non RTP NE DEVRONT PAS envoyer de blocs DLRR. Si une taille maximum en octets est incluse, tout bloc de rapport DLRR qui est envoyé NE DEVRA PAS excéder la taille spécifiée. Si les limitations de taille signifient qu'un envoyeur de bloc de rapport DLRR ne peut pas faire rapport en un seul bloc sur tous les participants auxquels il a reçu un bloc de rapport d'heure de référence de receveur, il DEVRAIT alors faire rapport sur les participants à la façon round robin à travers plusieurs intervalles de rapport.

La liste de paramètres d'attribut "rtcp-xr" PEUT être vide. Ceci est utile dans les cas où une application a besoin de signaler qu'elle comprend la signalisation SDP mais ne souhaite pas profiter elle-même de la fonction XR. Par exemple, une application dans une session contrôlée par SIP pourrait signaler qu'elle souhaite cesser d'utiliser tous les blocs XR en retirant tous les paramètres SDP applicables dans un message re-INVITE qu'elle envoie. Si les blocs XR ne sont pas du tout à utiliser depuis le début d'une session, il est RECOMMANDÉ que l'attribut "rtcp-xr" ne soit pas fourni du tout.

Lorsque un attribut "rtcp-xr" est présent, les participants NE DEVRAIENT PAS envoyer de blocs XR autres que ceux qui sont indiqués par les paramètres. Cela signifie que l'inclusion d'un attribut "rtcp-xr" sans aucun paramètre dit à un participant qu'il NE DEVRAIT PAS envoyer du tout de bloc XR. L'idée est de conserver la bande passante. C'est particulièrement important lorsque des paramètres collaboratifs sont appliqués à un gros groupe de diffusion groupée : l'envoi d'un bloc initiateur pourrait éventuellement déclencher des réponses de la part de tous les participants. Il y a cependant des contextes dans lesquels il y a du sens à envoyer un bloc XR en l'absence d'un paramètre signalant son utilisation. Par exemple, une application pourrait être conçue de façon à envoyer certains blocs de rapport sans négociation, tout en utilisant la signalisation SDP pour négocier l'utilisation des autres blocs.

## 5.2 Usage en offre/réponse

Dans le contexte offre/réponse [8], l'interprétation de la signalisation SDP pour les paquets XR dépend de l'attribut de direction qui est signalé : "recvonly", "sendrecv", ou "sendonly" [4]. Si aucun attribut de direction n'est fourni, on suppose alors "sendrecv". Ce paragraphe ne s'applique qu'aux flux de support en envoi individuel, sauf indication contraire. La discussion des paramètres unilatéraux est suivie de celle des paramètres collaboratifs.

Pour les offres de flux de support "sendonly" et "sendrecv" qui spécifient des paramètres d'attribut "rtcp-xr" unilatéraux, les réponses DEVRAIENT envoyer les blocs XR correspondants. Pour les offres "sendrecv", ceux qui répondent PEUVENT inclure l'attribut "rtcp-xr" dans sa réponse, et spécifier tout paramètre unilatéral afin de demander que l'offreur envoie les blocs XR correspondants. L'offreur DEVRAIT envoyer ces blocs.

Pour les offres de flux de support "recvonly", l'utilisation par l'offreur de l'attribut "rtcp-xr" en connexion avec des paramètres unilatéraux indique que l'offreur est capable d'envoyer les blocs XR correspondants. Si celui qui répond le fait avec un attribut "rtcp-xr", l'offreur DEVRAIT envoyer des blocs XR pour chaque paramètre unilatéral spécifié qui était dans son offre.

Pour les flux de support en diffusion groupée, l'inclusion d'un attribut "rtcp-xr" avec des paramètres unilatéraux signifie que chaque receveur du support DEVRAIT envoyer les blocs XR correspondants.

Une offre SDP avec un paramètre collaboratif déclare l'offreur capable de recevoir l'initiateur et répondant correspondants avec les réponses appropriées. Par exemple, une offre qui spécifie le paramètre "rcvr-rtt" signifie que l'offreur est prêt à recevoir des blocs de rapport Heure de référence du receveur et d'envoyer des blocs de rapport DLRR. Une offre d'un paramètre collaboratif signifie que celui qui répond PEUT envoyer l'initiateur, et, ayant reçu l'initiateur, l'offreur DEVRAIT envoyer les réponses.

Il y a des exceptions à la règle qu'un offreur de paramètre collaboratif devrait envoyer des réponses. Par exemple, le paramètre collaboratif peut spécifier un mode qui exclut l'offreur; ou des considérations de contrôle d'encombrement ou d'unité maximum de transmission peuvent militer contre la réponse de l'offreur.

En incluant un paramètre collaboratif dans une réponse, celui qui répond déclare sa capacité à recevoir des initiateurs et à envoyer des réponses. L'offreur PEUT alors envoyer des initiateurs, auxquels le répondant DEVRAIT répliquer par des réponses. Comme pour l'offre d'un paramètre collaboratif, il y a des exceptions à la règle que le répondant devrait répondre.

Lorsque il fait une offre SDP de paramètre collaboratif pour un flux de support en diffusion groupée, l'offreur DEVRAIT spécifier quels participants doivent répondre à un initiateur reçu. Un participant qui n'est pas spécifié NE DEVRAIT PAS envoyer de réponses. Autrement, de la bande passante indue pourrait être consommée. L'offre indique chaque participant spécifié qui DEVRAIT répondre si il reçoit un initiateur. Elle indique aussi qu'un participant spécifié PEUT envoyer un bloc initiateur.

Une réponse SDP pour un flux de support en diffusion groupée DEVRAIT inclure tous les paramètres collaboratifs qui sont présents dans l'offre et qui sont pris en charge par celui qui répond. Elle NE DEVRAIT PAS inclure de paramètre collaboratif qui serait absent de l'offre.

Si un participant reçoit une offre SDP et s'il comprend l'attribut "rtcp-xr" mais ne souhaite pas mettre en œuvre la fonction XR offerte, sa réponse DEVRAIT inclure un attribut "rtcp-xr" sans paramètre. Faisant ainsi, cette partie déclare qu'au minimum, il est capable de comprendre la signalisation.

### 5.3 Usage en dehors de offre/réponse

SDP peut être employé en dehors du contexte d'offre/réponse, par exemple pour des sessions multimédia qui sont annoncées par le protocole d'annonce de session (SAP) [15], ou pour des flux en direct du protocole de flux directs en temps réel (RTSP) [17]. Le modèle de signalisation est plus simple, car l'expéditeur ne négocie pas de paramètres, mais la fonctionnalité attendue de la spécification de l'attribut "rtcp-xr" est la même que dans l'offre/réponse.

Lorsque un paramètre unilatéral est spécifié pour l'attribut "rtcp-xr" associé à un flux de support, le receveur de ce flux DEVRAIT envoyer le bloc XR correspondant. Lorsque un paramètre collaboratif est spécifié, seuls les participants indiqués par la valeur de mode du paramètre collaboratif sont concernés. Chacun de ces participants qui reçoit un bloc initiateur DEVRAIT envoyer le bloc de réponse correspondant. Chacun de ces participants PEUT aussi envoyer des blocs initiateurs.

## 6. Considérations relatives à l'IANA

Le présent document définit un nouveau type de paquet RTCP, le type Rapport étendu (XR, *Extended Report*) au sein du registre existant de l'autorité d'allocation des numéros de l'Internet (IANA, *Internet Assigned Numbers Authority*) des types de paquets de contrôle de RTP RTCP. Le présent document définit aussi un nouveau registre de l'IANA : le registre des types de bloc XR RTCP. Au sein de ce nouveau registre, le présent document définit un ensemble initial de sept types de bloc et décrit comment les types restants sont à allouer.

De plus, le présent document définit un nouvel attribut SDP, "rtcp-xr", au sein du registre IANA existant des paramètres SDP. Il définit un nouveau registre de l'IANA, le registre des paramètres RTCP XR SDP, et un ensemble initial de six paramètres, et décrit comment des paramètres supplémentaires seront alloués.

### 6.1 Type de paquet XR

Le type de paquet XR défini par le présent document est enregistré auprès de l'IANA comme type de paquet 207 dans le registre des types de paquet (PT) de contrôle RTP RTCP.

### 6.2 Registre de type de bloc XR RTCP

Le présent document crée un registre IANA appelé Registre des types de bloc XR RTCP pour couvrir l'espace de noms du champ Type de bloc (BT) de rapport étendu spécifié à la Section 3. Le champ BT contient huit bits, permettant 256 valeurs. Le registre des types de bloc XR RTCP est à gérer par l'IANA conformément à la politique de spécification exigée de la RFC 2434 [7]. De futures spécifications DEVRAIENT attribuer des valeurs de type de bloc dans l'ordre numérique strict à la suite des valeurs attribuées dans le présent document :



<b>BT</b>	<b>nom</b>
1	bloc de rapport RLE de perte
2	bloc de rapport RLE dupliqué
3	bloc de rapport heure de réception de paquet
4	bloc de rapport heure de référence de receveur
5	bloc de rapport DLRR
6	bloc de rapport statistiques sommaires
7	bloc de rapport métriques VoIP

La valeur de BT 255 est réservée pour de futures extensions.

De plus, les futures spécifications DEVRAIENT éviter la valeur 0. Le faire facilite la vérification de validité des paquets, car un champ tout de zéros peut couramment se trouver dans des paquets mal formés.

Tout enregistrement DOIT contenir les informations suivantes :

- Informations de contact sur la personne qui effectue l'enregistrement, incluant au moins nom, adresse, et mël.
- Le format du type de bloc enregistré, cohérent avec le format de bloc de rapport étendu décrit à la Section 3.
- Une description de ce que le type de bloc représente et comment il devra être interprété, en détaillant ces informations pour chacun de ses champs.

### 6.3 Attribut SDP "rtcp-xr"

L'attribut SDP "rtcp-xr" défini dans la présent document est enregistré auprès de l'IANA dans le registre des paramètres SDP comme suit :

Attribut SDP	("att-field"):
Nom d'attribut :	rtcp-xr
Forme longue :	paramètres de rapport étendu de protocole de contrôle RTP
Type de nom :	att-field
Type d'attribut :	niveau session et support
Soumis à un jeu de caractères :	non
Objet :	voir la Section 5 de ce document
Référence :	le présent document
Valeurs :	voir ce document et les enregistrements ci-dessous

L'attribut a un champ de paramètre extensible et donc un registre est nécessaire pour ces paramètres. Le présent document crée un registre IANA appelé Registre des paramètres SDP XR RTCP. Il contient les six paramètres définis au paragraphe 5.1 : "pkt-loss-rle", "pkt-dup-rle", "pkt-rcpt-times", "stat-summary", "voip-metrics", et "recv-rtt".

Des paramètres supplémentaires sont à ajouter dans ce registre conformément à la politique de spécification exigée de la RFC 2434 [7]. Tout enregistrement DOIT contenir les informations suivantes :

- Informations de contact de celui qui effectue l'enregistrement, incluant au moins nom, adresse, et mël.
- Une définition en forme Backus-Naur augmentée (ABNF) [2] du paramètre, conformément à la définition "format-ext" du paragraphe 5.1.
- Une description de ce que le paramètre représente et comment il devra être interprété, normalement et en offre/réponse.

## 7. Considérations pour la sécurité

Le présent document étend le mécanisme de rapport RTCP. Les considérations de sécurité qui s'appliquent aux rapports RTCP (Appendice B de [9]) s'appliquent aussi aux rapports XR. Cette section détaille les considérations de sécurité supplémentaires qui s'appliquent aux extensions.

Les extensions introduisent des problèmes de confidentialité plus élevés. Les rapports RTCP standard contiennent un nombre limité de statistiques sommaires. Les informations contenues dans les rapports XR sont à la fois plus détaillés et plus étendus (couvrant un plus grand nombre de paramètres). Les blocs de rapport par paquet et le bloc de rapport Métrique VoIP en donnent des exemples.

Les informations par paquet contenues dans les blocs de rapport RLE de perte, RLE dupliqué, et heure de réception de paquet facilitent les conclusions pour la diffusion groupée des caractéristiques du réseau (MINC) [11]. De telles conclusions peuvent révéler la topologie globale d'une arborescence de distribution de diffusion groupée, ainsi que des

paramètres, tels que les taux de pertes et les délais, le long des chemins entre les points d'embranchement de cette arborescence. De telles informations pourraient être considérées comme sensibles par les administrateurs de système autonome.

Le bloc de rapport Métrique VoIP fournit des informations sur la qualité des appels vocaux sortants. Bien que de telles informations puissent être portées dans un format spécifique de l'application dans les sessions RTP standard, les rendre disponibles en format standard les rend ici plus disponibles à de potentiels espions.

Aucun nouveau mécanisme n'est introduit dans le présent document pour assurer la confidentialité. Les procédures de chiffrement, telles que celles suggérées pour un RTCP sécurisé (SRTCP) [12] au moment de la rédaction du présent document, peuvent être utilisées lorsque la confidentialité est un souci pour les hôtes d'extrémité. Étant donné que le trafic RTCP peut être chiffré par les hôtes d'extrémité, les systèmes autonomes doivent être préparés au fait que certains aspects de leur topologie de réseau peuvent être révélés.

Tout chiffrement ou filtrage des blocs de rapport XR entraîne une perte des informations de surveillance pour les tiers. Par exemple, un réseau qui établit un tunnel pour chiffrer les blocs de rapport VoIP refuse ces informations aux fournisseurs de service traversés par le tunnel. Les fournisseurs de service ne peuvent alors pas surveiller ou répondre de la qualité des appels VoIP qu'ils transportent, créant potentiellement des problèmes pour les utilisateurs du réseau. Par défaut, les paquets XR ne devraient pas être chiffrés ou filtrés.

Les extensions rendent aussi certaines attaques de déni de service plus faciles. Cela parce que la possibilité de créer des paquets RTCP beaucoup plus gros que la moyenne avec la capacité de rapport par paquet des blocs de rapport de RLE de perte, de RLE dupliqué et d'horodatage. À cause des mécanismes d'ajustement automatique de la bande passante dans RTCP, si certains participants à la session envoient de gros paquets RTCP, tous les participants verront leurs intervalles de rapport RTCP ralentis, ce qui signifie qu'il ne seront capables de faire rapport que moins fréquemment. Pour limiter les effets des gros paquets, même en l'absence d'attaque de déni de service, les applications DEVRAIENT mettre une limite supérieure à la taille des blocs de rapport XR qu'elles utilisent. Les techniques "d'amincissement" décrites au paragraphe 4.1 permettent que les blocs de rapport paquet par paquet respectent une limite de taille prédéfinie.

## A. Algorithmes

### A.1 Interprétation du numéro de séquence

C'est l'algorithme suggéré au paragraphe 4.1 pour garder trace du numéro de séquence d'un certain envoyeur. Il met en œuvre les pratiques de comptabilité exigées pour la génération de bloc de rapport RLE de pertes.

Cet algorithme garde trace de numéros de séquence de 16 bits en les traduisant dans un espace de numéro de séquence à 32 bits. Le premier paquet reçu d'une source est considéré comme étant arrivé en gros au milieu de cet espace. Chaque paquet qui suit est placé soit devant, soit derrière le précédent dans cet espace de 32 bits, selon que ce choix le place le plus près (ou, dans le cas d'égalité, quel choix n'exigerait pas un retour à zéro du numéro de séquence à 16 bits).

// Le numéro de séquence de référence est un numéro de séquence étendu qui sert de base pour déterminer si un nouveau numéro de séquence à 16 bits vient avant ou après dans l'espace de numéros de séquence à 32 bits.

```
u_int32 _src_ref_seq;
bool _uninitialized_src_ref_seq;
```

// Placer seq dans un espace de numéro de séquence à 32 bits sur la base d'une heuristique pour sa situation la plus probable.

```
u_int32 extend_seq(const u_int16 seq) {
u_int32 extended_seq, seq_a, seq_b, diff_a, diff_b;
si (_uninitialized_src_ref_seq) {
```

// C'est le premier numéro de séquence reçu. Le placer au milieu de l'espace étendu de numéros de séquence.

```
_src_ref_seq = seq | 0x80000000u;
_uninitialized_src_ref_seq = faux;
extended_seq = _src_ref_seq;
}
autrement {
```

// Les numéros de séquence précédents ont été reçus. Proposer deux candidats pour le numéro de séquence étendu : seq\_a est sans retour à zéro, seq\_b avec retour à zéro.

```

seq_a = seq | (_src_ref_seq & 0xFFFF0000u);
si (_src_ref_seq < seq_a) {
    seq_b = seq_a - 0x00010000u;
    diff_a = seq_a - _src_ref_seq;
    diff_b = _src_ref_seq - seq_b;
}
autrement {
    seq_b = seq_a + 0x00010000u;
    diff_a = _src_ref_seq - seq_a;
    diff_b = seq_b - _src_ref_seq;
}

```

// Choisir le candidat le plus proche. si ils sont également proches, le choix est un peu arbitraire : on choisit le candidat pour lequel le retour à zéro n'est pas nécessaire.

```

si (diff_a < diff_b) {
    extended_seq = seq_a;
}
autrement {
    extended_seq = seq_b;
}

```

// Régler le numéro de séquence de référence comme étant le numéro de séquence reçu le plus récemment.

```

_src_ref_seq = extended_seq;
}

```

// Retourner la meilleure hypothèse pour un numéro de séquence à 32 bits qui corresponde au numéro à 16 bits qui était donné.

```

retourner extended_seq;
}

```

## A.2 Exemple de calcul de perte de paquet de salve

C'est un algorithme pour mesurer les caractéristiques des salves pour le bloc de rapport de métrique VoIP (paragraphe 4.7). L'algorithme, dont la correction a été vérifiée à l'aide d'une mise en œuvre réelle, est reproduit d'après la spécification technique ETSI TS 101 329-5 [3]. L'algorithme, tel que décrit ici, l'emporte sur tout changement qui pourrait éventuellement être apporté à l'algorithme dans de futurs documents de l'ETSI.

Le moteur de cet algorithme est l'événement et son efficacité de calcul est donc forte.

Étant données les définitions d'états suivantes :

```

état 1 = un paquet est reçu durant un intervalle
état 2 = un paquet est reçu durant une salve
état 3 = un paquet est perdu durant une salve
état 4 = un paquet isolé est perdu durant un intervalle

```

Les variables "c" ci-dessous correspondent aux comptes de transition d'état, c'est-à-dire que c14 est la transition de l'état 1 à l'état 4. Il est possible de déduire un des comptes de transition d'état d'une paire avec une précision de 1, ce qui est généralement suffisant pour cette application.

"pqt" est le compte de paquets reçus depuis que le dernier paquet a été déclaré perdu ou éliminé, et "perdu" est le nombre de paquets perdus au sein de la salve actuelle. "paquet\_perdu" et "paquet\_éliminé" sont des variables booléennes qui indiquent si l'événement qui a pour résultat l'invocation de cette fonction était un paquet perdu ou éliminé.

```

si (paquet_perdu) {
    compte_perte++;
}
si (paquet_éliminé) {
    compte_éliminé++;
}
si (!paquet_perdu && !paquet_éliminé) {
    pqt++;
}

```

```

}
autrement {
  si(pqt ≥ gmin) {
    si(perdu == 1) {
      c14++;
    }
    autrement {
      c13++;
    }
    perdu = 1;
    c11 += pqt;
  }
  autrement {
    perdu++;
    si(pqt == 0) {
      c33++;
    }
    autrement {
      c23++;
      c22 += (pqt - 1);
    }
  }
  pqt = 0;
}

```

À chaque intervalle de rapport, les métriques de salve et d'intervalle peuvent être calculées comme suit.

```

// Calculer les comptes de transition supplémentaires.
c31 = c13;
c32 = c23;
ctotal = c11 + c14 + c13 + c22 + c23 + c31 + c32 + c33;

// Calculer les salves et les densités.
p32 = c32 / (c31 + c32 + c33);
si((c22 + c23) < 1) {
  p23 = 1;
}
autrement {
  p23 = 1 - c22/(c22 + c23);
}
densité_de_salve = 256 * p23 / (p23 + p32);
densité_d'intervalle = 256 * c14 / (c11 + c14);

// Calculer les durées de salve et d'intervalle en ms
m = duréeDeTrame_en_ms * tramesParPqtRTP ;
longueur_d'intervalle = (c11 + c14 + c13) * m / c13 ;
longueur_de_salve = ctotal * m / c13 - lintervalle ;

/* calculer les taux de perte et d'élimination */
taux_de_perte = 256 * compte_de_perte / ctotal ;
taux_d'élimination = 256 * compte_délimination / ctotal;

```

## Déclaration de droits de propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait 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'IETF au sujet des droits dans les documents en cours de normalisation et se rapportant aux normes figurent dans le BCP 11.

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 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, brevet ou applications de brevets, ou autres droits de propriété qui pourraient recouvrir la technologie qui pourrait être nécessaire pour mettre en œuvre la présente norme. Prière d'adresser les informations au directeur exécutif de l'IETF.

## Remerciements

Nous remercions les personnes suivantes : Colin Perkins, Steve Casner, et Henning Schulzrinne pour leurs conseils éclairés ; Sue Moon pour son aide au maintien de la collaboration entre les auteurs ; Mounir Benzaid pour avoir attiré notre attention sur les besoins de rapport de MLDA ; Dorgham Sisalem et Adam Wolisz pour nous avoir encouragés à incorporer les types de bloc MLDA ; et Jose Rey pour sa précieuse relecture de la section sur la signalisation SDP.

## Contributeurs

Les personnes suivantes sont les auteurs du présent document : Kevin Almeroth (UCSB), Ramon Caceres (IBM Research), Alan Clark (Telchemy), Robert G. Cole (JHU Applied Physics Laboratory), Nick Duffield (AT&T Labs-Research), Timur Friedman (Paris 6), Kaynam Hedayat (Brix Networks), Kamil Sarac (UT Dallas), Magnus Westerlund (Ericsson).

Les principales personnes à contacter pour ce qui concerne les blocs de rapport individuels décrits dans le présent document sont les suivantes :

<b>Paragraphe du bloc de rapport</b>	<b>principaux contributeurs</b>
4.1 Bloc de rapport RLE de perte	Friedman, Caceres, Duffield
4.2 Bloc de rapport RLE dupliqué	Friedman, Caceres, Duffield
4.3 Bloc de rapport Heure de réception de paquet	Friedman, Caceres, Duffield
4.4 Bloc de rapport Heure de référence de receveur	Friedman
4.5 Bloc de rapport DLRR	Friedman
4.6 Bloc de rapport Statistiques sommaires	Almeroth, Sarac
4.7 Bloc de rapport Métriques VoIP	Clark, Cole, Hedayat

La principale personne à contacter concernant la signalisation SDP décrite dans ce document est Magnus Westerlund.

## Références

### Références normatives

- [1] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", [RFC2119], BCP 14, mars 1997.
- [2] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", [RFC2234], novembre 1997. (*Obsolète, voir RFC5234*)
- [3] ETSI TS 101 329-5 V1.1.1, "Quality of Service (QoS) measurement methodologies", novembre 2000.
- [4] M. Handley et V. Jacobson, "SDP : [Protocole de description de session](#)", [RFC2327], avril 1998. (*Obsolète; voir RFC4566*)
- [5] R. Hovey, S. Bradner, "Les organisations impliquées dans le processus de normalisation de l'IETF", [RFC2028], octobre 1996. (*MàJ par RFC3668, RFC3979*) (BCP0011)
- [6] Recommendation UIT-T G.107, "The E-Model, a computational model for use in transmission planning", janvier 2003.
- [7] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", [RFC2434], BCP 26, octobre, 1998. (*Rendue obsolète par la RFC5226*)
- [8] J. Rosenberg et H. Schulzrinne, "[Modèle d'offre/réponse](#) avec le protocole de description de session (SDP)", [RFC3264], juin 2002.

- [9] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "[RTP : un protocole de transport pour les applications](#) en temps réel", [RFC3550], STD 64, juillet 2003.
- [10] TIA/EIA-810-A "Transmission Requirements for Narrowband Voice over IP et Voice over PCM Digital Wireline Telephones", décembre 2000.

### Références pour information

- [11] Adams, A., Bu, T., Caceres, R., Duffield, N.G., Friedman, T., Horowitz, J., Lo Presti, F., Moon, S.B., Paxson, V. et D. Towsley, "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior", IEEE Communications Magazine, mai 2000.
- [12] M. Baugher et autres, "Protocole de transport sécurisé en temps réel (SRTP)", [RFC3711], mars 2004.
- [13] Caceres, R., Duffield, N.G. et T. Friedman, "Impromptu measurement infrastructures using RTP", Proc. IEEE Infocom 2002.
- [14] Clark, A.D., "Modeling the Effects of Burst Packet Loss et Recency on Subjective Voice Quality", Proc. IP Telephony Workshop 2001.
- [15] M. Handley, C. Perkins, E. Whelan, "Protocole d'annonce de session (SAP)", [RFC2974], octobre 2000. (*Expérimentale*)
- [16] J. Reynolds, "[Numéros alloués](#) : la RFC 1700 est remplacée par une base de données en ligne", [RFC3232], janvier 2002.
- [17] H. Schulzrinne, A. Rao et R. Lanphier, "Protocole de [flux directs en temps réel](#) (RTSP)", [RFC2326], avril 1998.
- [18] Sisalem D. et A. Wolisz, "MLDA: A TCP-friendly Congestion Control Framework for Heterogeneous Multicast Environments", Proc. IWQoS 2000.

### Adresse des auteurs

Kevin Almeroth  
Department of Computer Science  
University of California  
Santa Barbara, CA 93106 USA  
mél : [almeroth@cs.ucsb.edu](mailto:almeroth@cs.ucsb.edu)

Ramon Caceres  
IBM Research  
19 Skyline Drive  
Hawthorne, NY 10532 USA  
mél : [caceres@watson.ibm.com](mailto:caceres@watson.ibm.com)

Alan Clark  
Telchemy Incorporated  
3360 Martins Farm Road, Suite 200  
Suwanee, GA 30024 USA  
mél : [alan@telchemy.com](mailto:alan@telchemy.com)

Robert G. Cole	Nick Duffield	Timur Friedman
Johns Hopkins Univ. Applied Physics Lab.	AT&T Labs-Research	Université Pierre et Marie Curie
MP2-S170	180 Park Avenue, P.O. Box 971	Laboratoire LiP6-CNRS
11100 Johns Hopkins Road	Florham Park, NJ 07932-0971 USA	8, rue du Capitaine Scott
Laurel, MD 20723-6099 USA	mél : <a href="mailto:duffield@research.att.com">duffield@research.att.com</a>	75015 PARIS France
mél : <a href="mailto:robert.cole@jhuapl.edu">robert.cole@jhuapl.edu</a>		mél : <a href="mailto:timur.friedman@lip6.fr">timur.friedman@lip6.fr</a>

Kaynam Hedayat  
Brix Networks  
285 Mill Road  
Chelmsford, MA 01824 USA  
mél : [khedayat@brixnet.com](mailto:khedayat@brixnet.com)

Kamil Sarac  
Department (ES 4.207)  
Eric Jonsson School of Engineering  
University of Texas at Dallas  
Richardson, TX 75083-0688 USA  
mél : [ksarac@utdallas.edu](mailto:ksarac@utdallas.edu)

Magnus Westerlund  
Ericsson Research  
Ericsson AB  
SE-164 80 Stockholm Sweden  
mél : [magnus.westerlund@ericsson.com](mailto:magnus.westerlund@ericsson.com)

### Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2003). Tous droits réservés.

Ce document et les traductions de celui-ci peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de droits de reproduction ci-dessus et ce paragraphe soit inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définis dans les processus de normes pour Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet ou ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et l'INTERNET SOCIETY et l'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.

**Remerciement**

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