

Groupe de travail Réseau

L. Barbato, Xiph

Request for Comments : 5215

Catégorie : Sur la voie de la normalisation

août 2008

Traduction Claude Brière de L'Isle

Format de charge utile RTP pour audio codé en Vorbis

Statut du présent mémoire

Le présent document spécifie un protocole Internet sur la voie de la 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 "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Le présent document décrit un format de charge utile RTP pour transporter l'audio codé en Vorbis. Il détaille le mécanisme d'encapsulation RTP pour les données Vorbis brutes et les mécanismes de livraison pour le modèle de probabilité du décodeur (auquel on se réfère comme à un livre de code) ainsi que les autres informations d'établissement.

Le présent mémoire inclut aussi les enregistrements de type de supports et les détails nécessaires à l'utilisation de Vorbis avec le protocole de description de session (SDP, *Session Description Protocol*).

Table des Matières

1. Introduction.....	1
1.1 Conformité et conventions du document.....	2
2. Format de charge utile.....	2
2.1 En-tête RTP.....	2
2.2 En-tête de charge utile.....	3
2.3 Données de charge utile.....	3
2.4 Exemple de paquet RTP.....	4
3. En-têtes de configuration.....	5
3.1 Transmission d'en-tête dans la bande.....	5
3.2 Transmission hors bande.....	6
3.3 Perte des en-têtes de configuration.....	7
4. En-têtes de commentaires.....	7
5. Mise en paquet de trame.....	8
5.1 Exemple de paquet Vorbis fragmenté.....	8
5.2 Perte de paquet.....	10
6. Considérations relatives à l'IANA.....	10
6.1 Considérations de l'IANA sur les en-têtes mis en paquets.....	10
7. Considérations relatives à SDP.....	11
7.1 Transposition des paramètres de type de support en SDP.....	11
7.2 Usage avec le modèle SDP d'offre/réponse.....	11
8. Contrôle d'encombrement.....	11
9. Exemple.....	12
9.1 Flux radio.....	12
10. Considérations sur la sécurité.....	12
11. Conditions de copie.....	12
12. Remerciements.....	12
13. Références.....	13
13.1 Références normatives.....	13
13.2 Références pour information.....	13
Adresse de l'auteur.....	13
Déclaration complète de droits de reproduction.....	14

1. Introduction

Vorbis est un codec audio d'objet général destiné à permettre une souplesse maximale de codage, permettant donc de s'adapter de façon compétitive à une gamme exceptionnellement large de débits binaires. À l'extrémité supérieure de

3. En-têtes de configuration

À la différence des codecs audio courants, Vorbis n'a pas de modèle de probabilité configuré statiquement. À la place, il prend tous les modèles de configuration de décodage entropique, de quantification de vecteur et de Huffman dans un bloc de données qui doit être transmis au décodeur avec les données compressées. Un décodeur exige aussi des informations précisant le nombre de canaux audio, les débits binaires, et informations similaires pour se configurer pour un flux de données compressées particulier. Ces deux blocs d'informations sont souvent appelés collectivement les "livres de code" pour un flux Vorbis, et sont inclus dans des paquets spéciaux "d'en-tête" au début des données compressées. De plus, la spécification [VORBIS] exige la présence d'un paquet d'en-tête de commentaire qui donne des métadonnées simples sur le flux, mais ces informations ne sont pas exigées pour décoder la séquence de trames.

Donc, ces deux paquets d'en-tête de livre de code doivent être reçus par le décodeur avant que des données audio puissent être interprétées. Ces exigences posent des problèmes dans RTP, qui est souvent utilisé sur des transports non fiables.

Comme ces informations doivent être transmises de façon fiable et, comme le flux RTP peut changer certaines données de configuration en cours de session, il y a différentes méthodes pour livrer ces données de configuration à un client, dans la bande et hors bande, qui sont détaillées ci-dessous. Afin d'établir un état initial pour l'application de client, la configuration DOIT être portée via le canal de signalisation utilisé pour établir la session. Un exemple de cette signalisation est SDP [RFC4566] avec le modèle d'offre/réponse [RFC3264]. Les changements de configuration PEUVENT être communiqués via un re-invite, portant un nouveau SDP, ou envoyés dans la bande dans le canal RTP. Les mises en œuvre DOIVENT prendre en charge une livraison dans la bande des livres de code mis à jour, et DEVRAIENT prendre en charge la mise à jour de livre de code hors bande en utilisant un nouveau fichier SDP. Les changements peuvent être dus à des livres de code différents ainsi qu'à des débits binaires différents du flux RTP.

Pour les flux non chaînés, la méthode de livraison de configuration recommandée est dans la configuration mise en paquet (paragraphe 3.1.1) dans le SDP, comme expliqué dans la transposition de paramètres de type de supports en SDP (paragraphe 7.1).

Le champ Ident de 24 bits est utilisé pour transposer la configuration qui va être utilisée pour décoder un paquet. Quand le champ Ident change, cela indique qu'un changement a eu lieu dans le flux. L'application de client DOIT avoir la configuration correcte à l'avance. Si le client détecte un changement dans la valeur de Ident et n'a pas cette information, il NE DOIT PAS décoder les données Vorbis brutes associées tant qu'il n'est pas allé chercher la configuration correcte.

3.1 Transmission d'en-tête dans la bande

La charge utile de configuration mise en paquet (paragraphe 3.1.1) est envoyée dans la bande avec les bits de type de paquet réglés au type de données Vorbis. Les clients DOIVENT être capables de traiter la fragmentation et la retransmission périodique des en-têtes de configuration [RFC4588]. La valeur de l'horodatage RTP DOIT refléter l'heure de transmission du premier paquet de données auquel cette configuration s'applique.

3.1.1 Configuration mise en paquet

Une configuration mise en paquet Vorbis est indiquée par le champ Type de données Vorbis à 1. Des trois en-têtes définis dans la spécification [VORBIS], Identification et Établissement DOIVENT être mis en paquet comme ils sont, tandis que l'en-tête Commentaires PEUT être remplacé par un en-tête factice.

La configuration mise en paquet mémorise les configurations de codec Xiph d'une façon générique : le premier champ mémorise le nombre des paquets qui suivent moins un (champ Compte) les suivants représentent la taille des en-têtes (champs Longueur) et les en-têtes suivent immédiatement la liste des champs Longueur. La taille du dernier en-tête est implicite.

Les champs Compte et Longueur sont codés en utilisant la logique suivante : les données sont dans l'ordre des octets du réseau et chaque octet a le bit de poids fort utilisé comme un fanion, et les 7 bits suivants sont utilisés pour mémoriser la valeur. Les 7 premiers bits de poids fort sont mémorisés dans le premier octet. Si il y a des bits restants, le bit fanion est réglé à 1 et les 7 bits suivants sont mémorisés dans l'octet suivant. Si il y a des bits restants, on règle le bit fanion à 1 et la même procédure est répétée. L'octet final a le bit fanion réglé à 0. Pour décoder, on itère simplement parmi les octets jusqu'à ce que le bit fanion soit réglé à 0. Pour chaque octet, les données sont ajoutées à la valeur accumulée multipliée par 128.

Les en-têtes sont mis en paquets dans le même ordre que présents dans Ogg [VORBIS] : Identification, Commentaire, Établissement.

L'étiquette de longueur de 2 octets définit la longueur des en-têtes mis en paquet comme la somme des longueurs de Configuration, Commentaires, et Établissement.



Figure 5 : Configuration mise en paquets

Le champ Ident est réglé à la valeur qui va être utilisée par les paquets de charge utile brute pour traiter cette configuration. Le type Fragment est réglé à 0 parce que le paquet porte la configuration Pleine mise en paquet. Le nombre de paquet est réglé à 1.

3.2 Transmission hors bande

La définition de paquet suivante DOIT être utilisée quand Configuration est dans le SDP.

3.2.1 En-têtes mis en paquets

Comme mentionné ci-dessus, le vecteur de livraison RECOMMANDÉ pour les données de configuration Vorbis est via une méthode de restitution qui peut être effectuée en utilisant un protocole de transport fiable. Comme il n'est pas obligé que les en-têtes RTP soient présents pour cette méthode de livraison, la structure des données de configuration est légèrement différente. L'en-tête mis en paquet commence par un champ de compte de 32 bits (dans l'ordre des octets du réseau) avec les détails du nombre d'en-têtes mis en paquet qui sont contenus dans le bouquet. Le diagramme suivant montre la charge utile d'en-tête mis en paquet pour chaque flux Vorbis enchaîné.

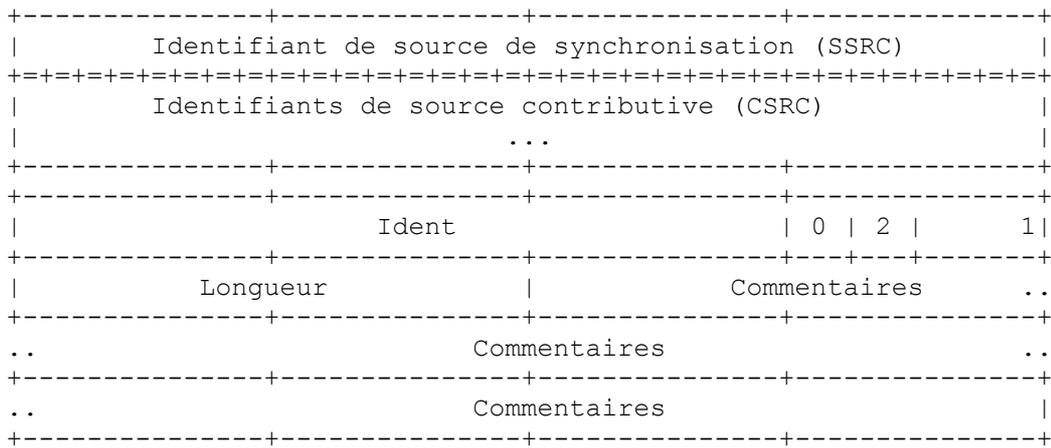


Figure 8 : Paquet de commentaires

Le champ Longueur de 2 octets est nécessaire car ce paquet pourrait être fragmenté.

5. Mise en paquet de trame

Chaque charge utile RTP contient soit un fragment de paquet Vorbis, soit un nombre entier de paquets Vorbis complets (jusqu'à un maximum de 15 paquets, car le nombre de paquets est défini par une valeur de 4 bits).

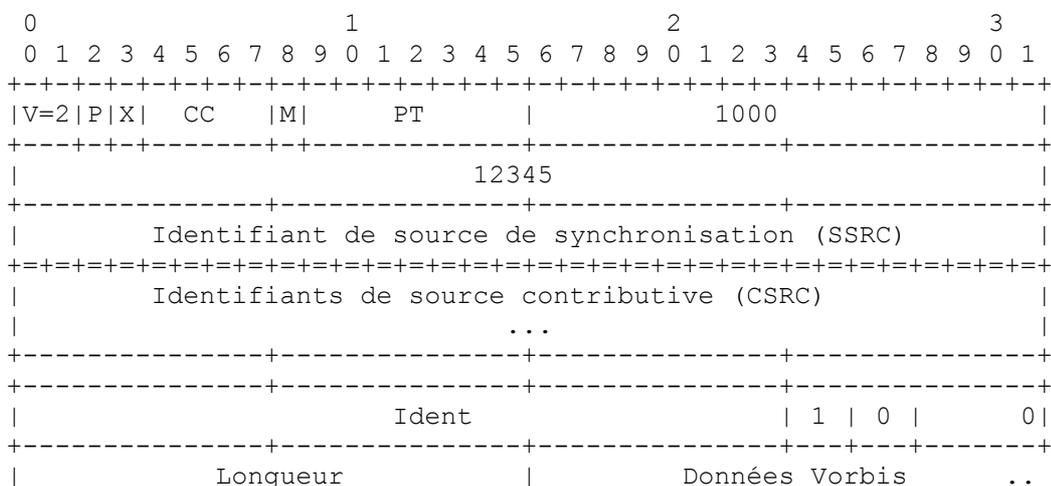
Tout paquet de données Vorbis qui fait moins que la MTU de chemin DEVRAIT être groupé dans la charge utile RTP avec autant de paquets Vorbis qui vont y tenir, jusqu'à un maximum de 15, sauf quand un tel groupement excéderait la latence de transmission désirée de l'application. La MTU de chemin est détaillée dans les [RFC1191] et [RFC1981].

Un paquet fragmenté a un zéro dans les quatre derniers bits de l'en-tête de la charge utile. Le premier fragment va établir le type de fragment à 1. Chaque fragment après le premier va régler le type de fragment à 2 dans l'en-tête de charge utile. Les fragments consécutifs DOIVENT être envoyés sans autre charge utile envoyée entre le premier et le dernier fragment. La charge utile RTP qui contient le dernier fragment du paquet Vorbis va avoir le type de fragment réglé à 3. Pour conserver la séquence correcte pour la réception du paquet fragmenté, le champ Horodatage des paquets fragmentés DOIT être le même que celui du premier paquet envoyé, avec le numéro de séquence incrémenté normalement pour les charges utiles RTP suivantes ; cela va affecter la mesure de gigue RTCP. Le champ Longueur montre la longueur du fragment.

5.1 Exemple de paquet Vorbis fragmenté

Voici un exemple d'un paquet Vorbis fragmenté partagé en trois charges utiles RTP. Chacune d'elles contient les en-têtes RTP standard ainsi que les en-têtes Vorbis de 4 octets.

Paquet 1 :



```

+-----+-----+-----+-----+
..                               Données Vorbis   |
+-----+-----+-----+-----+

```

Figure 9 : Exemple de paquet fragmenté (paquet 1)

Dans cette charge utile, le numéro de séquence initial est 1000 et l'horodatage est 12345. Le type de fragment est réglé à 1, le champ Nombre de paquets est réglé à 0, et comme la charge utile est de données Vorbis brutes, le champ VDT est à 0.

Paquet 2 :

```

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
+-----+-----+-----+-----+
|V=2|P|X| CC |M| PT | 1001 |
+-----+-----+-----+
|                               12345 |
+-----+-----+-----+
| Identifiant de source de synchronisation (SSRC) |
+-----+-----+-----+
| Identifiants de source contributive (CSRC) |
| ... |
+-----+-----+-----+
| Ident | 2 | 0 | 0 |
+-----+-----+-----+
| Longueur | Données Vorbis .. |
+-----+-----+-----+
..                               Données Vorbis |
+-----+-----+-----+

```

Figure 10 : Exemple de paquet fragmenté (paquet 2)

Le champ Type de fragment est réglé à 2, et le champ Nombre de paquets est réglé à 0. Pour les grands fragments Vorbis, il peut y avoir plusieurs de ces types de charge utile. La taille maximum de paquet DEVRAIT n'être pas supérieure à la MTU du chemin, incluant tous les en-têtes RTP et de charge utile. Le numéro de séquence a été incrémenté de un, mais le champ horodatage reste le même que celui de la charge utile initiale.

Paquet 3 :

```

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
+-----+-----+-----+-----+
|V=2|P|X| CC |M| PT | 1002 |
+-----+-----+-----+
|                               12345 |
+-----+-----+-----+
| Identifiant de source de synchronisation (SSRC) |
+-----+-----+-----+
| Identifiants de source contributive (CSRC) |
| ... |
+-----+-----+-----+
| Ident | 3 | 0 | 0 |
+-----+-----+-----+
| Longueur | Données Vorbis .. |
+-----+-----+-----+
..                               Données Vorbis |
+-----+-----+-----+

```

Figure 11 : Exemple de paquet fragmenté (paquet 3)

Ceci est la dernière charge utile de fragment Vorbis. Le type de fragment est réglé à 3 et le compte de paquets reste réglé à 0. Comme dans les charges utiles précédentes, l'horodatage reste réglé à l'horodatage de la première charge utile dans la séquence et le numéro de séquence a été incrémenté.

5.2 Perte de paquet

Comme il n'y a pas de correction d'erreur au sein du flux Vorbis, la perte d'un paquet résulte en une perte du signal. La perte de paquet est plus un problème pour les paquets fragmentés Vorbis que le client va devoir traiter avec le type de fragment. En cas de perte de fragments, le client DOIT éliminer tous les fragments Vorbis restants et décoder le paquet incomplet. Si on utilise l'exemple de paquet Vorbis fragmenté ci-dessus et si la première charge utile RTP est perdue, le client DOIT détecter que la prochaine charge utile RTP a le champ Compte de paquets réglé à 0 et le type de fragment 2 et DOIT l'éliminer. La prochaine charge utile RTP, qui est le paquet fragmenté final, DOIT être éliminée de la même manière. Si la charge utile RTP manquante est la dernière, les deux fragments reçus vont être gardés et le paquet Vorbis incomplet sera décodé.

La perte de tout fragment Configuration va résulter en la perte du paquet de configuration complet avec le résultat précisé dans la section "Perte des en-têtes de configuration" (paragraphe 3.3).

6. Considérations relatives à l'IANA

Nom de type : audio

Nom de sous type : vorbis

Paramètres exigés :

rate : indique le taux d'horloge d'horodatage RTP comme décrit dans le profil RTP pour conférences audio et vidéo avec contrôle minimal [RFC3551].

channels : indique le nombre de canaux audio comme décrit dans le profil RTP pour conférences audio et vidéo avec contrôle minimal [RFC3551].

configuration : représentation en base64 [RFC4648] des en-têtes de mise en paquet (paragraphe 3.2.1).

Considérations de codage : ce type de support est tramé et contient des données binaires.

Considérations de sécurité : voir la Section 10 de la RFC 5215.

Considérations d'interopérabilité : aucune

Spécification publiée : RFC 5215, spécification Ogg Vorbis I : établissement de codec et décodage de paquet. Disponible sur le site Xiph de la Toile, <http://xiph.org/>

Applications qui utilisent ce type de support : flux audio et outils de conférence.

Information supplémentaires : aucune

Personne et adresse de messagerie à contacter pour plus d'informations : Luca Barbato: <lu_zero@gentoo.org>

Groupe de travail IETF Transport audio/vidéo

Usage prévu : COMMUN

Restriction d'usage : ce type de support dépend du tramage RTP, est n'est donc défini que pour le transfert via RTP [RFC3550].

Auteur : Luca Barbato

Contrôleur des changements : groupe de travail IETF AVT sur délégation de l'IESG.

6.1 Considérations de l'IANA sur les en-têtes mis en paquets

Les considérations suivantes se réfèrent aux en-têtes mis en paquets de configuration partagés (paragraphe 3.2.1) utilisés dans la RFC 5215.

Nom de type : audio

Nom de sous type : vorbis-config

Paramètres exigés : aucun

Paramètres facultatifs : aucun

Considérations de codage : ce type de support contient des données binaires.

Considérations de sécurité : voir la Section 10 de la RFC 5215.

Considérations d'interopérabilité : aucune

Spécification publiée : RFC 5215

Applications qui utilisent ce type de support : audi codé en Vorbis, données de configuration.

Information supplémentaires : aucune

Personne et adresse de messagerie à contacter pour plus d'informations : Luca Barbato: <lu_zero@gentoo.org>

Groupe de travail IETF Transport audio/vidéo

Usage prévu : COMMUN

Restriction d'usage : ce type de support ne dépend pas du transport.

Auteur : Luca Barbato

Contrôleur des changements : groupe de travail IETF AVT sur délégation de l'IESG.

7. Considérations relatives à SDP

Les paragraphes qui suivent définissent la transposition des paramètres décrits dans la section des considérations relatives à l'IANA et leur usage dans le modèle d'offre/réponse [RFC3264]. Pour être rétro compatible, la mise en œuvre DOIT ignorer les paramètres inconnus.

7.1 Transposition des paramètres de type de support en SDP

Les informations portées dans la spécification de type de support ont une transposition spécifique en les champs du protocole de description de session (SDP, *Session Description Protocol*) [RFC4566], qui est couramment utilisé pour décrire les sessions RTP. Quand SDP est utilisé pour spécifier les sessions, la transposition est comme suit :

- o Le nom de type ("audio") va dans SDP "m=" comme nom de support.
- o Le nom de sous type ("vorbis") va dans SDP "a=rtpmap" comme nom de codage.
- o Le paramètre "rate" va aussi dans "a=rtpmap" comme débit d'horloge.
- o Le paramètre "channels" va aussi dans "a=rtpmap" comme compte de canaux.
- o Les paramètres obligatoires "configuration" DOIVENT être inclus dans l'attribut SDP "a=fmtp".

Si le flux comporte des fichiers Vorbis chaînés et si tous sont connus à l'avance, le paquet de configuration pour chaque fichier DEVRAIT être passé au client en utilisant l'attribut "configuration".

La valeur d'accès est spécifiée par l'application de serveur liée à l'adresse spécifiée dans la ligne c=. La valeur de compte de canaux spécifiée dans l'attribut rtpmap DEVRAIT correspondre au flux Vorbis en cours ou devrait être considérée comme le nombre maximum de canaux à attendre. Le débit d'horloge d'horodatage DOIT être un multiple du taux d'échantillonnage ; un numéro de charge utile différent DOIT être utilisé si le débit d'horloge change. La charge utile de configuration délivre les informations exactes, donc les informations SDP DEVRAIENT être considérées comme un conseil. Un exemple est donné ci-dessous.

7.1.1 Exemple de SDP

L'exemple suivant montre un seul flux SDP de base. Le premier paquet de configuration est dans le SDP ; les autres configurations pourraient être récupérées à tout moment à partir des URI fournis. La chaîne de configuration en base64 [RFC4648] suivante va à la ligne dans cet exemple à cause des limitations de longueur de ligne des RFC.

```
c=IN IP4 192.0.2.1
m=audio RTP/AVP 98
a=rtpmap:98 vorbis/44100/2
a=fmtp:98 configuration=AAAAAZ2f4g9NAh4aAXZvcmJpcwA...;
```

Noter que les noms de format de charge utile (codage) sont généralement donnés en majuscules. Les sous types de type de support sont généralement donnés en minuscules. Dans les deux cas, ces noms ne sont pas sensibles à la casse. De même, les noms de paramètre sont insensibles à la casse dans les types de support et dans la transposition par défaut en attribut SDP a=fmtp. La ligne a=fmtp est une seule ligne, même si elle est montrée comme plusieurs lignes dans ce document.

7.2 Usage avec le modèle SDP d'offre/réponse

Il n'y a pas de paramètre négociable. Tous sont déclaratifs.

8. Contrôle d'encombrement

Les considérations générales de contrôle d'encombrement pour le transport des données RTP s'appliquent aussi à l'audio Vorbis sur RTP. Voir la spécification de RTP [RFC3550] et tout profil RTP applicable (par exemple, la [RFC3551]). Les données audio peuvent être codées en utilisant une gamme de débits binaires différents, de sorte qu'il est possible d'adapter

la bande passante du réseau en ajustant le débit binaire du codeur en temps réel ou en ayant plusieurs copies du contenu codé à des débits binaires différents.

9. Exemple

L'exemple suivant montre un schéma d'utilisation courant qui PEUT être appliqué dans une telle situation. Le principal objectif de cette section est de mieux expliquer l'usage des vecteurs de transmission.

9.1 Flux radio

C'est une des situations les plus courantes : il y a un seul serveur qui diffuse du contenu en diffusion groupée, et les clients peuvent commencer une session à un moment aléatoire. Le contenu lui-même pourrait être un mélange de flux directs (comme de la voix d'une webjockey) et des flux mémorisés (comme la musique qu'elle joue).

Dans cette situation, on ne sait pas à l'avance combien de livres de code on va utiliser. Les clients peuvent se joindre à tout moment et les utilisateurs s'attendent à commencer l'écoute du contenu dans peu de temps.

Lorsque il se joint, le client reçoit la configuration actuelle nécessaire pour décoder le flux actuel à l'intérieur de SDP de sorte que le décodage va commencer immédiatement après.

Quand le contenu du flux change, la nouvelle configuration est envoyée dans la bande avant le flux actuel, et la configuration qui doit être envoyée dans le SDP est mise à jour. Comme la méthode dans la bande n'est pas fiable, une solution de repli hors bande est fournie.

Le client peut choisir d'aller chercher la configuration auprès de la source de remplacement aussitôt qu'il découvre qu'un paquet de configuration est perdu dans la bande, ou d'utiliser une retransmission sélective [RFC3611] si le serveur prend en charge cette caractéristique.

Une optimisation côté serveur serait de garder une liste hachée des configurations par session, ce qui évite de les mettre toutes en paquets et d'envoyer la même configuration avec des étiquettes Ident différentes.

Une optimisation côté client serait de tenir une liste d'étiquettes des configurations par session et de ne pas traiter les paquets de configuration qu'il connaît déjà.

10. Considérations sur la sécurité

Les paquets RTP qui utilisent ce format de charge utile sont soumis aux considérations de sécurité discutées dans la spécification RTP [RFC3550], la spécification base64 [RFC4648], et la spécification de la syntaxe générique d'URI [RFC3986]. Entre autres considérations, cela implique que la confidentialité du flux de supports est réalisée par le chiffrement. Parce que la compression de données utilisée avec ce format de charge utile est appliquée de bout en bout, le chiffrement peut être effectué sur les données compressées.

11. Conditions de copie

L'auteur accorde aux tiers le droit irrévocable de copier, utiliser, et distribuer ce travail, avec ou sans modification, dans tout support, sans redevance, pourvu que, sauf si une permission spéciale est accordée, les travaux modifiés redistribués ne contiennent pas d'informations fallacieuses sur l'auteur, la version, le nom du travail ou son appartenance.

12. Remerciements

Le présent document est une continuation des documents suivants :
Moffitt, J., "Format de charge utile RTP pour audio codé en Vorbis", février 2001.
Kerr, R., "Format de charge utile RTP pour audio codé en Vorbis", décembre 2004.

La déclaration de type de support est une continuation du document de B. Short, "The audio/rtp-vorbis MIME Type", janvier 2008.

Merci au groupe de travail AVT, aux communautés Vorbis/Xiph.Org Foundation incluant Steve Casner, Aaron Colwell, Ross Finlayson, Fluendo, Ramon Garcia, Pascal Hennequin, Ralph Giles, Tor-Einar Jarnbjo, Colin Law, John Lazzaro, Jack Moffitt, Christopher Montgomery, Colin Perkins, Barry Short, Mike Smith, Phil Kerr, Michael Sparks, Magnus Westerlund, David Barrett, Silvia Pfeiffer, Stefan Ehmann, Gianni Ceccarelli, et Alessandro Salvatori. Merci au groupe LScube, en particulier à Federico Ridolfo, Francesco Varano, Giampaolo Mancini, Dario Gallucci, et Juan Carlos De Martin.

13. Références

13.1 Références normatives

- [RFC1191] J. Mogul et S. Deering, "[Découverte de la MTU](#) de chemin", novembre 1990.
- [RFC1981] J. McCann, S. Deering, J. Mogul, "Découverte de la [MTU de chemin pour IP version 6](#)", août 1996. (D.S. ; Remplacé par [RFC8201], STD87)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par RFC8174)
- [RFC3264] J. Rosenberg et H. Schulzrinne, "[Modèle d'offre/réponse](#) avec le protocole de description de session (SDP)", juin 2002. (P.S. ; MàJ par RFC8843, 9143)
- [RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "[RTP : un protocole de transport pour les applications en temps réel](#)", STD 64, juillet 2003. (MàJ par RFC7164, RFC7160, RFC8083, RFC8108, RFC8860)
- [RFC3551] H. Schulzrinne et S. Casner, "[Profil RTP pour conférences audio](#) et vidéo avec contrôle minimal", STD 65, juillet 2003. (MàJ par RFC8860)
- [RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiant de ressource uniforme](#) (URI) : Syntaxe générique", STD 66, janvier 2005. (P.S. ; MàJ par RFC8820)
- [RFC4566] M. Handley, V. Jacobson et C. Perkins, "SDP : [Protocole de description de session](#)", juillet 2006. (P.S. ; remplacée par RFC8866)
- [RFC4648] S. Josefsson, "[Codages de données Base16, Base32 et Base64](#)", octobre 2006. (Remplace RFC3548) (P.S.)
- [VORBIS] "Ogg Vorbis I specification: Codec setup and packet decode". Disponible sur le site de Xiph : http://xiph.org/vorbis/doc/Vorbis_I_spec.html.

13.2 Références pour information

- [LIBVORBIS] "libvorbis", disponible sur le site dédié <http://vorbis.com/>.
- [RFC3533] S. Pfeiffer, "Format d'encapsulation Ogg, version 0", mai 2003. (Information)
- [RFC3611] T. Friedman, R. Caceres et A. Clark, éditeurs, "[Rapports étendus du protocole](#) de contrôle de RTP (RTCP XR)", novembre 2003. (P.S.)
- [RFC4588] J. Rey et autres, "[Format de charge utile](#) de retransmission RTP", juillet 2006. (P.S.)

Adresse de l'auteur

Luca Barbato
Xiph.Org Foundation
mél : lu_zero@gentoo.org
URI : <http://xiph.org/>

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2008).

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

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

Propriété intellectuelle

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

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

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