

Groupe de travail Réseau
Request pour Comments : 2435
 RFC rendue obsolète : 2035
 Catégorie : Sur la voie de la normalisation
 Traduction Claude Brière de L'Isle

L. Berc, Digital Equipment Corporation
 W. Fenner, Xerox PARC
 R. Frederick, Xerox PARC
 S. McCanne, Lawrence Berkeley Laboratory
 P. Stewart, Xerox PARC
 octobre 1998

Format de charge utile RTP pour vidéo compressée en JPEG

Statut de ce mémoire

Le présent document spécifie un protocole en cours de normalisation de l'Internet 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.

Notice de copyright

Copyright (C) The Internet Society (1998). Tous droits réservés.

Résumé

Le présent mémoire décrit le format de charge utile RTP pour les flux vidéo JPEG. Le format de paquet est optimisé pour les flux vidéo en temps réel où les paramètres de codec changent rarement d'une trame à l'autre.

Le présent document a été produit par le groupe de travail Transport audio-vidéo au sein de l'équipe d'ingénierie de l'Internet (IETF). Des commentaires sont sollicités et devraient être adressés à la liste de diffusion du groupe de travail à rem-conf@es.net et/ou aux auteurs.

Changements par rapport à la RFC 2035

La plus grande partie de ce mémoire est identique à la RFC 2035. Les changements au protocole sont résumés à l'Appendice D.

Mots clés

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

Table des matières

1. Introduction.....	2
2. JPEG sur RTP.....	2
3. Format de paquet RTP/JPEG.....	3
3.1 En-tête JPEG.....	3
4. Discussion.....	5
4.1 Champ Type.....	5
4.2 Champ Q.....	6
4.3 Fragmentation et réassemblage.....	6
4.4 Marqueurs de redémarrage.....	7
5. Considérations sur la sécurité.....	7
6. Adresse des auteurs.....	8
7. Références.....	8
Appendice A.....	9
Appendice B.....	10
Appendice C.....	13
Appendice D.....	16
Déclaration complète de droits de reproduction.....	16

1. Introduction

Les normes du groupe conjoint d'experts photographiques (JPEG, *Joint Photographic Experts Group*) [T.81], [JPEG91], [JPEG93] définissent une famille d'algorithmes de compression pour les images fixes en tonalité continue. Cette norme de compression d'image fixe peut être appliquée à la vidéo en compressant chaque trame de vidéo comme une image fixe indépendante et en les transmettant en série. La vidéo codée de cette façon est souvent appelée "JPEG animé".

On donne d'abord une vue d'ensemble de JPEG et on décrit ensuite le sous ensemble spécifique de JPEG qui est pris en charge dans RTP et le mécanisme par lequel les trames JPEG sont portées comme des charges utiles RTP.

La norme JPEG définit quatre modes de fonctionnement : le mode DCT séquentiel, le mode DCT progressif, le mode sans perte, et le mode hiérarchique. Selon le mode, l'image est représentée en un ou plusieurs passages. Chaque passage (appelé une trame dans la norme JPEG) est de plus coupé en un ou plusieurs examens. Dans chaque examen, il y a de une à quatre composantes, qui représentent les trois composantes d'un signal de couleur (par exemple, "rouge, vert, et bleu", ou un signal de luminance et deux de chrominance). Ces composantes peuvent être codées comme des examens séparés ou entrelacés dans un seul examen.

Chaque trame et chaque examen est précédé d'un en-tête contenant les définitions facultatives des paramètres de compression comme les tableaux de quantification et les tableaux de codage de Huffman. Les en-têtes et les paramètres facultatifs sont identifiés par des "marqueurs" et comportent un segment marqueur ; chaque examen apparaît comme un flux binaire à entropie codée dans deux segments marqueurs. Les marqueurs sont alignés sur les limites d'octets et (en général) ne peuvent pas apparaître dans le segment à entropie codée, ce qui permet que les limites d'examen soient déterminées sans analyser le flux binaire.

Les données compressées sont représentées dans un des trois formats suivants : format d'échange, format abrégé, format de spécification de tableau. Le format d'échange contient les définitions pour tous les tableaux utilisés par les segments à entropie codée, tandis que le format abrégé pourrait en omettre certaines en supposant qu'elles ont été définies hors bande ou par une image "précédente".

La norme JPEG ne définit pas la signification ou le format des composants qui constituent l'image. Des attributs comme l'espace de couleur et le ratio d'aspect de pixel doivent être spécifiés hors bande par rapport au flux binaire JPEG. Le format d'échange de fichier JPEG (JFIF, *JPEG File Interchange Format*) [JPEG-FIF] est une norme de fait qui donne des informations supplémentaires en utilisant un segment marqueur d'application (APP0). Note qu'un fichier JFIF est simplement une image en format d'échange JPEG avec le segment APP0. Dans le cas de vidéo, des paramètres supplémentaires doivent être définis hors bande (par exemple, taux de trames, entrelacé ou non entrelacé, etc.).

Bien que la norme JPEG fournisse un riche ensemble d'algorithmes pour une compression souple, il n'y a pas eu de mises en œuvre de matériels efficaces en coût de la norme complète. La plupart des codecs vidéo JPEG mettent plutôt en œuvre seulement un sous ensemble du mode de fonctionnement DCT séquentiel. Normalement, les segments marqueurs sont interprétés dans un logiciel (qui "reprogramme" le matériel) et un seul examen à entropie codée, entrelacé, est présenté au matériel, représenté dans l'espace de couleur YUV.

L'examen contient une séquence ordonnée d'unités codées minimales (MCU, *Minimum Coded Unit*) qui sont le plus petit groupe de données d'image codées dans un flux binaire JPEG. Chaque MCU définit les données d'image pour un petit bloc rectangulaire de l'image de sortie.

Les marqueurs de redémarrage dans les données JPEG notent un point où le décodeur devrait rétablir son état. Comme défini par JPEG, les marqueurs de redémarrage sont le seul type de marqueur qui peut apparaître incorporé dans le segment à entropie codée, et ils peuvent seulement apparaître sur une frontière de MCU. Un "intervalle de redémarrage" est défini comme un bloc de données contenant un marqueur de redémarrage suivi par un nombre fixé de MCU. Une exception est faite pour le premier intervalle de redémarrage dans chaque trame, qui omet le marqueur de redémarrage initial et commence juste avec les données de MCU. Quand ces marqueurs sont utilisés, chaque trame est composée d'un nombre fixé d'intervalles de redémarrage dos à dos.

2. JPEG sur RTP

Pour maximiser l'interopérabilité entre les codecs fondés sur le matériel, on suppose le mode de fonctionnement DCT séquentiel [T.81], Annexe F, et on restreint l'ensemble de "codes de type" prédéfini RTP/JPEG (défini ci-dessous) aux

images à un seul examen, entrelacées. Bien que ce soit plus restrictif que même le JPEG de base, de nombreuses mises en œuvre de matériel sont en deçà de la spécification de base (par exemple, la plupart des matériels ne peuvent pas décoder les examens non entrelacés).

En pratique, la plupart des données de spécification de tableau changent rarement d'une trame à l'autre dans un seul flux vidéo. Donc, les données RTP/JPEG sont représentées en format abrégé, tous les tableaux étant omis du flux binaire lorsque possible. Chaque trame commence immédiatement avec l'examen (le seul) à entropie codée. Les informations qui seraient autrement dans les en-têtes de trame et d'examen sont représentées entièrement dans l'en-tête RTP/JPEG (défini ci-dessous) qui se tient entre l'en-tête RTP et la charge utile JPEG.

Bien que des paramètres comme les tableaux Huffman et l'espace de couleur doivent probablement rester fixés pour la durée de vie du flux vidéo, les autres paramètres devrait pouvoir varier, notamment les tableaux de quantification et la taille d'image (par exemple, pour mettre en œuvre la transmission à taux adaptatif ou pour permettre à un utilisateur d'ajuster le "niveau de qualité" ou la résolution manuellement). Donc, les champs explicites dans l'en-tête RTP/JPEG sont alloués pour représenter ces informations. Comme seulement un petit ensemble de tableaux de quantification est normalement utilisé, on code l'ensemble entier de tableaux de quantification dans un petit champ d'entier. Les tableaux de quantification personnalisés sont traités en utilisant une gamme spéciale de valeurs dans ce champ, et ensuite en plaçant le tableau avant le début de la charge utile JPEG. La largeur et la hauteur d'image sont codées explicitement.

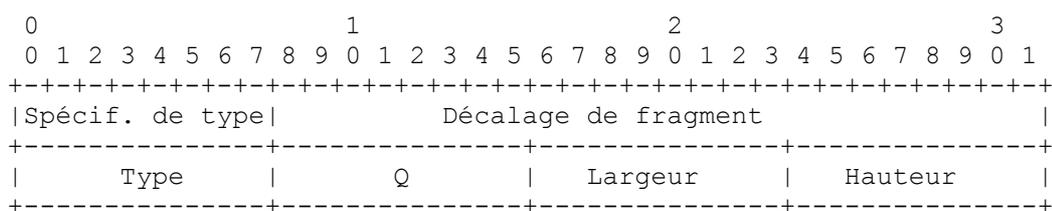
Parce que les trames JPEG sont normalement plus grandes que la taille maximum de paquet du réseau sous-jacent, les trames doivent souvent être fragmentées en plusieurs paquets. Une approche est de permettre à la couche réseau en dessous de RTP (par exemple, IP) d'effectuer la fragmentation. Cependant, cela empêche de contrôler le taux du flux de paquets résultant ou la livraison partielle en présence de pertes, et des trames peuvent être plus grandes que la longueur maximum de réassemblage de la couche réseau (voir plus d'informations dans [FRAG]). Pour éviter ces limitations, RTP/JPEG définit un schéma simple de fragmentation et réassemblage au niveau RTP.

3. Format de paquet RTP/JPEG

L'horodatage RTP est en unités de 90 000 Hz. Le même horodatage DOIT apparaître dans chaque fragment d'une trame. Le bit marqueur RTP DOIT être établi dans le dernier paquet d'une trame.

3.1 En-tête JPEG

Chaque paquet contient un en-tête JPEG spécial qui suit immédiatement l'en-tête RTP. Les 8 premiers octets de cet en-tête, appelés "en-tête JPEG principal", sont comme suit :



Tous les champs dans cet en-tête sauf le champ Décalage de fragment DOIVENT rester les mêmes dans tous les paquets qui correspondent à la même trame JPEG.

Un en-tête Marqueur de redémarrage et/ou Tableau de quantification peut suivre cet en-tête, selon les valeurs des champs Type et Q.

3.1.1 Spécifique du type : 8 bits

L'interprétation dépend de la valeur du champ Type. Si aucune interprétation n'est spécifiée, ce champ DOIT être mis à zéro à l'émission et ignoré à réception.

3.1.2 Décalage de fragment : 24 bits

Le décalage de fragment est le décalage en octets du paquet en cours dans les données de la trame JPEG. Cette valeur est

codée dans l'ordre des octets du réseau (octet de poids fort en premier). Le décalage de fragment plus la longueur des données de la charge utile dans le paquet NE DOIT PAS excéder 2^24 octets.

3.1.3 Type : 8 bits

Le champ Type spécifie les information qui seraient autrement présentes dans une spécification de tableau JPEG abrégé ainsi que les paramètres supplémentaires de style JFIF non définis par JPEG. Les types 0 à 63 sont réservés comme des transpositions fixées bien connues, à définir par le présent document et ses futures révisions. Les types 64 à 127 sont les mêmes que les types de 0 à 63, sauf que les marqueurs de redémarrage sont présents dans les données JPEG et qu'un en-tête Marqueur de redémarrage apparaît immédiatement après l'en-tête JPEG principal. Les types 128 à 255 sont à définir de façon dynamique par un protocole d'établissement de session (qui sort du domaine d'application de ce document).

3.1.4 Q : 8 bits

Le champ Q définit les tableaux de quantification pour cette trame. Les valeurs de Q de 0 à 127 indiquent que les tableaux de quantification sont calculés en utilisant un algorithme déterminé par le champ Type (voir ci-dessous). Les valeurs de Q de 128 à 255 indiquent qu'un en-tête Tableau de quantification apparaît après l'en-tête JPEG principal (et l'en-tête Marqueur de redémarrage, si présent) dans le premier paquet de la trame (décalage de fragment de 0). Cet en-tête peut être utilisé pour spécifier explicitement dans la bande les tableaux de quantification.

3.1.5 Largeur : 8 bits

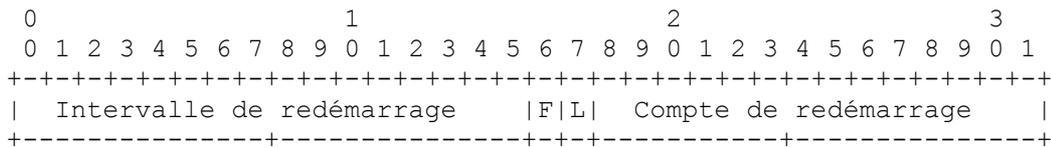
Ce champ code la largeur de l'image en multiples de 8 pixels (par exemple, une largeur de 40 note une image large de 320 pixels). La largeur maximum est de 2040 pixels.

3.1.6 Hauteur : 8 bits

Ce champ code la hauteur de l'image en multiples de 8 pixels (par exemple, une hauteur de 30 note une image haute de 240 pixels). Lors du codagee de vidéo entrelacée, c'est la hauteur d'un champ Vidéo, car les champs sont codés individuellement en JPEG. La hauteur maximum est de 2040 pixels.

3.1.7 En-tête Marqueur de redémarrage

Cet en-tête DOIT être présent immédiatement après l'en-tête JPEG principal quand on utilise les types de 64 à 127. Il fournit les informations supplémentaires requises pour décoder de façon appropriée un flux de données contenant des marqueurs de redémarrage.



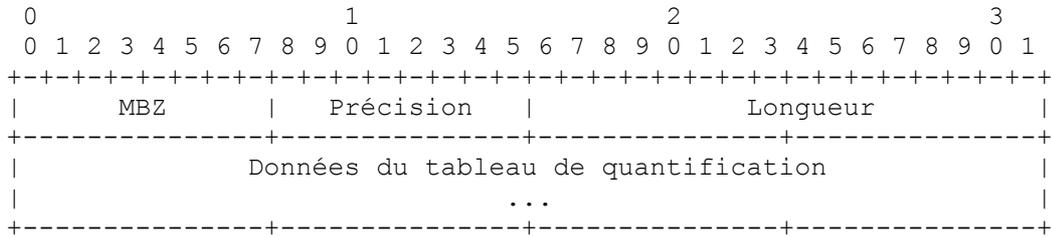
Le champ Intervalle de redémarrage spécifie le nombre de MCU qui apparaissent entre marqueurs de redémarrage. Il est identique à la valeur de 16 bits qui apparaîtrait dans le segment marqueur DRI d'un en-tête JFIF. Cette valeur NE DOIT PAS être zéro.

Si les intervalles de redémarrage dans une trame ne sont pas d'alignement garanti avec les limites de paquet, les bits F (premier) et L (dernier) DOIVENT être réglés à 1 et le compte de redémarrage DOIT être réglé à 0x3FFF. Cela indique qu'un receveur DOIT réassembler la trame entière avant de la décoder.

Pour prendre en charge le décodage partiel de trame, la trame est coupée en "tronçons" dont chacun contient un nombre entier d'intervalles de redémarrage. Le champ Compte de redémarrage contient la position du premier intervalle de redémarrage dans le "tronçon" en cours afin que les receveurs sachent à quelle partie de la trame correspondent ces données. Une valeur d'intervalle de redémarrage DEVRAIT être choisie de façon à permettre qu'un "tronçon" tienne entièrement dans un seul paquet. Dans ce cas, les bits F et L du paquet sont tous deux réglés à 1. Cependant, si un tronçon doit être partagé en plusieurs paquets, le bit F va être réglé à 1 dans le premier paquet du tronçon (et seulement celui là) et le bit L va être réglé à 1 dans le dernier paquet du tronçon (et seulement celui-là).

3.1.8 En-tête Tableau de quantification

Cet en-tête DOIT être présent après l'en-tête JPEG principal (et après l'en-tête Marqueur de redémarrage, si présent) quand on utilise les valeurs de Q de 128 à 255. Il fournit un moyen de spécifier les tableaux de quantification associés à cette valeur de Q dans la bande.



Le champ Longueur est réglé à la longueur en octets des données du tableau de quantification à suivre. Le champ Longueur PEUT être réglé à zéro pour indiquer qu'aucune donnée de tableau de quantification n'est incluse dans cette trame. Voir plus d'informations au paragraphe 4.2. Si le champ Longueur dans un paquet reçu fait plus que le nombre d'octets restants, le paquet DOIT être éliminé.

Quand des données de tableau sont incluses, le nombre de tableaux présents dépend du champ Type JPEG. Par exemple, le type 0 utilise deux tableaux (un pour le composant luminance et un partagé par les composants de chrominance). Chaque tableau est une matrice de 64 valeurs données en zig-zag, identique au format utilisé dans un segment de marqueur JFIF DQT.

Pour chaque tableau de quantification présent, un bit dans le champ Précision spécifie la taille des coefficients dans ce tableau. Si le bit est zéro, les coefficients sont de 8 bits, donnant une longueur de tableau de 64 octets. Si le bit est un, les coefficients sont de 16 bits pour une longueur de tableau de 128 octets. Pour les tableaux de 16 bits, les coefficients sont présentés dans l'ordre des octets du réseau. Le bit de droite dans le champ Précision (le bit 15 dans le diagramme ci-dessus) correspond au premier tableau et chaque tableau supplémentaire utilise le prochain bit sur la gauche. Les bits au delà de ceux qui correspondent aux tableaux nécessaires pour le type utilisé DOIVENT être ignorés.

Pour les valeurs de Q de 128 à 254, la transposition de valeur Q en données de tableau de quantification DOIT être statique, c'est-à-dire, les receveurs ont la garantie qu'ils ont seulement besoin de lire les données du tableau une fois afin de décoder correctement les trames envoyées avec cette valeur Q. Une valeur Q de 255 note que la transposition de tableau de quantification est dynamique et peut changer à chaque trame. Les décodeurs NE DOIVENT PAS dépendre d'une version antérieure des tableaux, et doivent recharger ces tableaux à chaque trame. Les paquets NE DOIVENT PAS contenir Q = 255 et Longueur = 0.

3.1.9 Charge utile JPEG

Les données qui suivent les en-têtes RTP/JPEG sont un segment à entropie codée consistant en un seul examen. L'en-tête d'examen n'est pas présent et est déduit de l'en-tête RTP/JPEG. L'examen se termine implicitement (c'est-à-dire, au point auquel l'image est complètement analysée) ou explicitement avec un marqueur EOI. L'examen peut être bourré à une longueur arbitraire avec des octets indéfinis. (Certains codecs matériels existants génèrent des lignes supplémentaire au bas d'une trame vidéo et la suppression de ces lignes exigerait un passage de décodage Huffman sur les données.)

Le code de type détermine si les marqueurs de redémarrage sont présents. Si un type prend en charge les marqueurs de redémarrage, le paquet DOIT contenir une valeur d'intervalle de redémarrage non zéro dans un en-tête Marqueur de redémarrage et les marqueurs de redémarrage DOIVENT apparaître sur des limites alignées sur l'octet commençant par un 0xFF entre les MCU à cet intervalle. Des octets 0xFF supplémentaires PEUVENT apparaître entre les intervalles de redémarrage. Ceci peut être utilisé dans le processus de mise en paquets pour aligner les données sur quelque chose comme une limite de mot pour une copie plus efficace. Les marqueurs de redémarrage NE DOIVENT PAS apparaître ailleurs que dans la charge utile JPEG. Les types qui ne prennent pas en charge les marqueurs de redémarrage NE DOIVENT PAS contenir de marqueurs de redémarrage dans la charge utile JPEG. Tous les paquets DOIVENT contenir un octet "bourré" 0x00 suivant tout vrai octet 0xFF généré par le codeur d'entropie ([T.81] paragraphe B.1.1.5).

4. Discussion

4.1 Champ Type

Le champ Type définit les paramètres de spécification de tableau abrégée et des styles JFIF supplémentaires non définis dans JPEG, car ils ne sont pas présents dans le corps des données JPEG transmises.

Trois gammes du champ Type sont actuellement définies. Les types 0 à 63 sont réservés comme des transpositions fixées bien connues qui sont définis par le présent document et ses futures révisions. Les types 64 à 127 sont les mêmes que les types 0 à 63, sauf que les marqueurs de redémarrage sont présents dans les données JPEG et qu'un en-tête Marqueur de redémarrage apparaît immédiatement à la suite de l'en-tête JPEG principal. Les types 128 à 255 ont toute liberté d'être définis dynamiquement par un protocole d'établissement de session (qui sort du domaine de ce document).

Dans le premier groupe de transpositions fixées, les types 0 et 1 sont actuellement définis, ainsi que les types correspondants 64 et 65 qui indiquent la présence de marqueurs de redémarrage. Ils correspondent à une spécification abrégée de tableau indiquant le mode "DCT séquentiel de base", des échantillons de 8 bits, des pixels carrés, trois composants dans l'espace de couleur YUV, les tableaux Huffman standard comme défini dans [T.81], Annexe K.3, et un seul examen entrelacé avec un sélecteur de composant d'examen qui indique les composants 1, 2, et 3 dans cet ordre. Les plans de couleur Y, U, et V correspondent aux numéros de composant 1, 2, et 3, respectivement. Le composant 1 (c'est-à-dire, le plan de luminance) utilise le numéro de tableau Huffman 0 et le numéro de tableau de quantification 0 (défini plus loin) et les composants 2 et 3 (c'est-à-dire, les plans de chrominance) utilisent le numéro de tableau Huffman 1 et le numéro de tableau de quantification 1 (défini plus loin).

Les numéros de type 2 à 5 sont réservés et NE DEVRAIENT PAS être utilisés. Les applications qui se fondent sur les versions antérieures de ce document (la RFC 2035) devraient être mises à jour pour indiquer la présence de marqueurs de redémarrage avec le type 64 ou 65 et l'en-tête Marqueur de redémarrage.

Les deux types RTP/JPEG actuellement définis sont décrits ci-dessous :

Types	Composant	Facteur d'échantillon		Numéro de tableau de quantification
		horizontal	vertical	
0, 64	1 (Y)	2	1	0
	2 (U)	1	1	1
	3 (V)	1	1	1
1, 65	1 (Y)	2	2	0
	2 (U)	1	1	1
	3 (V)	1	1	1

Ces facteurs d'échantillonnage indiquent que les composants de chrominance de type 0 vidéo sont échantillonnés horizontalement par 2 (souvent appelé 4:2:2) tandis que les composants de chrominance de type 1 vidéo sont échantillonnés horizontalement et verticalement par 2 (souvent appelé 4:2:0).

Les types 0 et 1 peuvent être utilisés pour porter les données d'image aussi bien examinées progressivement que entrelacées. Ceci est codé en utilisant le champ spécifique de type dans l'en-tête JPEG principal. Les valeurs suivantes sont définies :

- 0 : l'image est examinée progressivement. Sur un écran d'ordinateur, elle peut être affichée telle qu'elle à la largeur et hauteur spécifiées.
- 1 : l'image est un champ impair d'un signal vidéo entrelacé. La hauteur spécifiée dans l'en-tête JPEG principal est la moitié de la hauteur de l'image affichée entière. Ce champ devrait être désentrelacé avec le champ qui le suit afin que les lignes provenant de chaque image s'alternent. Les lignes correspondantes provenant du champ pair devraient apparaître juste au-dessus des mêmes lignes provenant du champ impair.
- 2 : l'image est un champ pair d'un signal vidéo entrelacé.
- 3 : l'image est un seul champ provenant d'un signal vidéo entrelacé, mais il devrait être affiché en pleine trame comme si il avait été reçu comme les champs à la fois pairs et impairs de la trame. Sur un écran d'ordinateur, chaque ligne de

l'image devrait être affichés deux fois, doublant la hauteur de l'image. L'Appendice B contient le code source en langage C pour transformer les paramètres de l'en-tête RTP/JPEG dans la trame JPEG et examiner les en-têtes qui sont absents des données charge utile.

4.2 Champ Q

Pour les types JPEG 0 et 1 (et leurs types correspondants 64 et 65) les valeurs de Q entre 1 et 99 inclus sont définies comme suit. Les autres valeurs inférieures à 128 sont réservées. Les types supplémentaires sont invités à utiliser cette définition si applicable.

Les deux types JPEG 0 et 1 exigent deux tableaux de quantification. Ces tableaux sont calculés comme suit. Pour $1 \leq Q \leq 99$, la formule du groupe JPEG indépendant [JPEG-codec] est utilisée pour produire un facteur d'adaptation S de :

$$\begin{aligned} S &= 5000 / Q && \text{pour } 1 \leq Q \leq 50 \\ &= 200 - 2 * Q && \text{pour } 51 \leq Q \leq 99 \end{aligned}$$

Cette valeur est alors utilisée pour adapter les tableaux K.1 et K.2 de [T.81] (en saturant chaque valeur à 8 bits) pour donner les numéros de tableau de quantification 0 et 1, respectivement. Le code source en langage C est fourni à l'Appendice A pour calculer ces tableaux.

Pour les valeurs de Q de 128 à 255, des tableaux de quantification définis dynamiquement sont utilisés. Ces tableaux peuvent être spécifiés dans la bande ou hors bande par un protocole d'établissement de session, mais l'en-tête Tableau de quantification DOIT être présent dans le premier paquet de chaque trame. Quand les tableaux sont spécifiés hors bande, ils peuvent être omis du paquet en réglant le champ Longueur dans cet en-tête à 0.

Quand les tableaux de quantification sont envoyés dans la bande, ils n'ont pas besoin d'être envoyés avec chaque trame. Comme dans le cas hors bande, les trames qui ne contiennent pas de tableau auront un en-tête Tableau de quantification avec un champ Longueur de 0. Alors que cela diminue les frais généraux de l'inclusion des tableaux, les nouveaux receveurs ne vont pas être capables de décoder correctement les trames entre le moment du démarrage et celui où ils reçoivent les tableaux.

4.3 Fragmentation et réassemblage

Comme les trames JPEG peuvent être grandes, elles doivent souvent être fragmentées. Les trames DEVRAIENT être fragmentées en paquets d'une manière qui évite la fragmentation à un niveau inférieur. Si on désire la prise en charge du décodage partiel, les trames DEVRAIENT être fragmentées de telle façon que chaque paquet contienne un nombre entier d'intervalles de redémarrage (voir ci-dessous).

Chaque paquet qui constitue une seule trame DOIT avoir le même horodatage, et le bit Marqueur RTP DOIT être établi dans le dernier paquet d'une trame. Le champ Décalage de fragment de chaque paquet est réglé au décalage d'octet de ses données de charge dans la trame d'origine. Les paquets qui constituent une trame DEVRAIENT être envoyés à la suite et les fragments qu'ils contiennent NE DOIVENT PAS se chevaucher.

Une trame entière peut être identifiée comme une séquence de paquets commençant par un paquet qui a un décalage de fragment de zéro et se termine par un paquet qui a le bit Marqueur RTP établi. Les paquets manquants peuvent être détectés par les numéros de séquence RTP ou par le décalage de fragment et les longueurs de chaque paquet. Le réassemblage pourrait être effectué sans le champ de décalage (c'est-à-dire, en utilisant seulement le bit Marqueur RTP et les numéros de séquence) mais une mise en œuvre efficace à une seule copie ne serait autrement pas possible en présence de paquets dans le désordre. De plus, si le dernier paquet de la trame précédente (contenant le bit Marqueur) a été éliminé, un receveur pourrait alors ne pas toujours détecter que la trame en cours est entièrement intacte.

4.4 Marqueurs de redémarrage

Les marqueurs de redémarrage indiquent un point dans le flux JPEG auquel le décodeur Huffman et les prédicteurs DC sont réinitialisés, permettant le début d'un décodage partiel à ce point. Pour tirer pleinement parti de cela, un décodeur doit cependant savoir quelles MCU d'une trame codent un intervalle de redémarrage particulier. Bien que la spécification JPEG originale fournisse un petit champ de numéro de séquence dans les marqueurs de redémarrage pour cet objet, il n'est pas assez grand pour traiter correctement la perte d'un paquet entier de données à une taille normale de MTU du réseau. L'en-tête Marqueur de redémarrage RTP/JPEG contient les informations supplémentaires nécessaires pour faire cela.

La taille des intervalles de redémarrage DEVRAIT être choisie pour permettre toujours qu'un nombre entier d'intervalles de redémarrage tiennent dans un seul paquet. Cela va garantir que les paquets peuvent être décodés indépendamment les uns des autres. Si un intervalle de redémarrage se trouve être plus grand qu'un paquet, les bits F et L dans l'en-tête Marqueur de redémarrage peuvent être utilisés pour le fragmenter, mais l'ensemble résultant de paquets doit être reçu en entier par le décodeur pour que cet intervalle de redémarrage soit décodé correctement.

Une fois qu'un décodeur a reçu soit un seul paquet avec les deux bits F et L établis, soit une séquence contiguë de paquets (sur la base du numéro de séquence RTP) qui commence par un bit F et se termine par un bit L, il peut commencer à décoder. La position de la MCU au début des données peut être déterminée en multipliant la valeur du compte de redémarrage par la valeur de l'intervalle de redémarrage. Un paquet (ou groupe de paquets comme identifié par les bits F et L) peut contenir tout nombre d'intervalles de redémarrage consécutifs.

Pour s'accommoder des codeurs qui génèrent des trames contenant des marqueurs de redémarrage mais ne peuvent pas fragmenter les données de cette manière, le champ Compte de redémarrage peut être réglé à 0x3FFF avec les deux bits F et L réglés à 1. Cela indique aux décodeurs que la trame entière doit être réassemblée avant de la décoder.

5. Considérations sur la sécurité

Les paquets RTP qui utilisent le format de charge utile défini dans la présente spécification sont soumis aux considérations de sécurité discutées dans la spécification RTP [RFC1889], et de tout profil RTP approprié (par exemple [RFC1890]). Cela implique que la confidentialité des flux de supports est réalisée par le chiffrement. Parce que la compression des données utilisée avec ce format de charge utile est appliquée de bout en bout, le chiffrement peut être effectué après la compression afin qu'il n'y ait pas de conflit entre les deux opérations.

Une menace potentielle de déni de service existe pour les codages de données qui utilisent des techniques de compression qui ont une charge de calcul non uniforme à l'extrémité de réception. L'attaquant peut injecter dans le flux des datagrammes pathologiques qui sont complexes à décoder et causent la surcharge du receveur. Cependant, ce codage ne présente aucune non uniformité significative.

Une autre menace potentielle de déni de service existe avec le mécanisme de fragmentation présenté ici. Les receveurs devraient être prêts à limiter la quantité totale de données associée à l'assemblage des trames reçues afin d'éviter un épuisement de ressources.

Comme avec tout protocole fondé sur IP, dans certaines circonstances un receveur peut être surchargé simplement par la réception de trop de paquets, désirés ou non. L'authentification de la couche réseau peut être utilisée pour éliminer les paquets provenant de sources non désirées, mais le coût de traitement de l'authentification elle-même peut être trop élevé. Dans un environnement de diffusion groupée, l'élagage de sources spécifiques va être mis en œuvre dans une future version de IGMP [RFC2236] et dans les protocoles d'acheminement de diffusion groupée pour permettre à un receveur de choisir à quelles sources il est permis de le joindre.

Un passage en revue de la sécurité de ce format de charge utile n'a trouvé aucune considération supplémentaire au delà de celles de la spécification RTP.

6. Adresse des auteurs

Lance M. Berc
Systems Research Center
Digital Equipment Corporation
130 Lytton Ave
Palo Alto CA 94301
USA
téléphone : +1 650 853 2100
mél : berc@pa.dec.com

William C. Fenner
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
USA
téléphone : +1 650 812 4816
mél : fenner@parc.xerox.com

Ron Frederick
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
USA
téléphone : +1 650 812 4459
mél : frederick@parc.xerox.com

Steven McCanne
University of California at Berkeley

Paul Stewart
Xerox PARC

Electrical Engineering et Computer Science
 633 Soda Hall
 Berkeley, CA 94720
 USA
 téléphone : +1 510 642 0865
 mél : mccanne@cs.berkeley.edu

3333 Coyote Hill Road
 Palo Alto, CA 94304
 USA
 téléphone : +1 650 812 4821
 mél : stewart@parc.xerox.com

7. Références

- [FRAG] Kent C., and J. Mogul, "Fragmentation Considered Harmful", Proceedings of the ACM SIGCOMM '87 Workshop on Frontiers dans Computer Communications Technology, août 1987.
- [JPEG91] William B. Pennebaker, Joan L. Mitchell, "JPEG: Still Image Data Compression Standard", Van Nostrand Reinhold, 1993.
- [JPEG93] Gregory K. Wallace, "The JPEG Sill Picture Compression Standard", Communications of the ACM, avril 1991, Vol 34, No. 1, pp. 31-44.
- [JPEG-FIF] "The JPEG File Interchange Format". Produit par C-Cube Microsystems, Inc., et disponible à <ftp://ftp.uu.net/graphics/jpeg/jfif.ps.gz>.
- [JPEG-codec] Tom Lane and al., "The Independent JPEG Group software JPEG codec". Code source disponible à <ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6a.tar.gz>.
- [RFC1889] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : protocole de transport pour applications en temps réel", janvier 1996. (*Obsolète, voir RFC3550* STD64)
- [RFC1890] H. Schulzrinne, "Profil RTP pour conférences audio et vidéo avec contrôle minimal", janvier 1996. (*Obsolète, voir RFC3551*) (P.S.)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (*MàJ par RFC8174*)
- [RFC2236] W. Fenner, "Protocole de gestion de groupe Internet, version 2", novembre 1997. (*Remplacée par RFC3376*)
- [T.81] Recommandation UIT-T T.81 (ISO DIS 10918-1), "Compression numérique et codage des images fixes en tonalité continue (JPEG)".

Appendice A

Le code suivant peut être utilisé pour créer un tableau de quantification à partir d'un facteur Q :

/* * Tableau K.1 de la spécification JPEG */

```
static const int jpeg_luma_quantizer[64] = {
    16, 11, 10, 16, 24, 40, 51, 61,
    12, 12, 14, 19, 26, 58, 60, 55,
    14, 13, 16, 24, 40, 57, 69, 56,
    14, 17, 22, 29, 51, 87, 80, 62,
    18, 22, 37, 56, 68, 109, 103, 77,
    24, 35, 55, 64, 81, 104, 113, 92,
    49, 64, 78, 87, 103, 121, 120, 101,
    72, 92, 95, 98, 112, 100, 103, 99
};
```

/* * Tableau K.2 de la spécification JPEG */


```

    0x29, 0x2a, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39,
    0x3a, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49,
    0x4a, 0x53, 0x54, 0x55, 0x56, 0x57, 0x58, 0x59,
    0x5a, 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69,
    0x6a, 0x73, 0x74, 0x75, 0x76, 0x77, 0x78, 0x79,
    0x7a, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89,
    0x8a, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98,
    0x99, 0x9a, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7,
    0xa8, 0xa9, 0xaa, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6,
    0xb7, 0xb8, 0xb9, 0xba, 0xc2, 0xc3, 0xc4, 0xc5,
    0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xd2, 0xd3, 0xd4,
    0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xe1, 0xe2,
    0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea,
    0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8,
    0xf9, 0xfa,
};

u_char chm_dc_codelens[] = { 0, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, };

u_char chm_dc_symbols[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, };

u_char chm_ac_codelens[] = { 0, 2, 1, 2, 4, 4, 3, 4, 7, 5, 4, 4, 0, 1, 2, 0x77, };

u_char chm_ac_symbols[] = {
    0x00, 0x01, 0x02, 0x03, 0x11, 0x04, 0x05, 0x21,
    0x31, 0x06, 0x12, 0x41, 0x51, 0x07, 0x61, 0x71,
    0x13, 0x22, 0x32, 0x81, 0x08, 0x14, 0x42, 0x91,
    0xa1, 0xb1, 0xc1, 0x09, 0x23, 0x33, 0x52, 0xf0,
    0x15, 0x62, 0x72, 0xd1, 0x0a, 0x16, 0x24, 0x34,
    0xe1, 0x25, 0xf1, 0x17, 0x18, 0x19, 0x1a, 0x26,
    0x27, 0x28, 0x29, 0x2a, 0x35, 0x36, 0x37, 0x38,
    0x39, 0x3a, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48,
    0x49, 0x4a, 0x53, 0x54, 0x55, 0x56, 0x57, 0x58,
    0x59, 0x5a, 0x63, 0x64, 0x65, 0x66, 0x67, 0x68,
    0x69, 0x6a, 0x73, 0x74, 0x75, 0x76, 0x77, 0x78,
    0x79, 0x7a, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87,
    0x88, 0x89, 0x8a, 0x92, 0x93, 0x94, 0x95, 0x96,
    0x97, 0x98, 0x99, 0x9a, 0xa2, 0xa3, 0xa4, 0xa5,
    0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xb2, 0xb3, 0xb4,
    0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xc2, 0xc3,
    0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xd2,
    0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda,
    0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9,
    0xea, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8,
    0xf9, 0xfa,
};

u_char *
MakeQuantHeader(u_char *p, u_char *qt, int tableNo)
{
    *p++ = 0xff;
    *p++ = 0xdb;          /* DQT */
    *p++ = 0;            /* bit de poids fort de la longueur */
    *p++ = 67;           /* bit de moindre poids de la longueur */
    *p++ = tableNo;
    memcpy(p, qt, 64);
    return (p + 64);
}

u_char *
MakeHuffmanHeader(u_char *p, u_char *codelens, int ncodes, u_char *symbols, int nsymbols, int tableNo, int tableClass)

```

```

{
    *p++ = 0xff;
    *p++ = 0xc4;           /* DHT */
    *p++ = 0;             /* bit de poids fort de la longueur */
    *p++ = 3 + ncodes + nsymbols; /* bit de moindre poids de la longueur */
    *p++ = (tableClass << 4) | tableNo;
    memcpy(p, codelens, ncodes);
    p += ncodes;
    memcpy(p, symbols, nsymbols);
    p += nsymbols;
    return (p);
}

```

u_char *

```

MakeDRIHeader(u_char *p, u_short dri) {
    *p++ = 0xff;
    *p++ = 0xdd;         /* DRI */
    *p++ = 0x0;         /* bit de poids fort de la longueur */
    *p++ = 4;           /* bit de moindre poids de la longueur*/
    *p++ = dri >> 8;    /* bit de poids fort de dri */
    *p++ = dri & 0xff;  /* bit de moindre poids de dri */
    return (p);
}

```

/*

* Arguments:

* type, largeur, hauteur : comme fourni dans l'en-tête RTP/JPEG

* lqt, cqt : tableaux de quantification soit dérivés du champ Q en utilisant MakeTables() soit comme spécifié en 4.2.

* dri : intervalle de redémarrage dans les MCU, ou 0 si il n'y a pas de redémarrage.

* p : pointeur sur la zone de retour

* Valeur de retour : longueur des en-têtes générés. Génère des en-têtes de trame et d'examen qui peuvent être ajoutés devant la charge utile de données RTP/JPEG pour produire une image JPEG compressée en format d'échange (sauf pour des déchets possibles en queue et en l'absence d'un marqueur EOI pour terminer l'examen).

*/

```

int MakeHeaders(u_char *p, int type, int w, int h, u_char *lqt, u_char *cqt, u_short dri)

```

```

{

```

```

    u_char *start = p;

```

/* convertit de blocs en pixels */

```

    w <<= 3;

```

```

    h <<= 3;

```

```

    *p++ = 0xff;

```

```

    *p++ = 0xd8;           /* SOI */

```

```

    p = MakeQuantHeader(p, lqt, 0);

```

```

    p = MakeQuantHeader(p, cqt, 1);

```

```

    si (dri != 0)

```

```

        p = MakeDRIHeader(p, dri);

```

```

    *p++ = 0xff;

```

```

    *p++ = 0xc0;         /* SOF */

```

```

    *p++ = 0;           /* bit de poids fort de la longueur */

```

```

    *p++ = 17;         /* bit de moindre poids de la longueur */

```

```

    *p++ = 8;          /* précision de 8 bits */

```

```

    *p++ = h >> 8;     /* bit de poids fort de la hauteur*/

```

```

    *p++ = h;          /* bit de moindre poids de la hauteur */

```

```

    *p++ = w >> 8;     /* bit de poids fort de la largeur*/

```

```

    *p++ = w;          /* bit de moindre poids de la largeur */

```

```

*p++ = 3;          /* nombre de composants */
*p++ = 0;          /* composant 0 */
si (type == 0)
    *p++ = 0x21;   /* hsamp = 2, vsamp = 1 */
autrement
*p++ = 0x22;      /* hsamp = 2, vsamp = 2 */
*p++ = 0;         /* tableau de quantification 0 */
*p++ = 1;         /* composant 1 */
*p++ = 0x11;     /* hsamp = 1, vsamp = 1 */
*p++ = 1;         /* tableau de quantification 1 */
*p++ = 2;         /* composant 2 */
*p++ = 0x11;     /* hsamp = 1, vsamp = 1 */
*p++ = 1;         /* tableau de quantification 1 */
p = MakeHuffmanHeader(p, lum_dc_codelens,
    sizeof(lum_dc_codelens),
    lum_dc_symbols,
    sizeof(lum_dc_symbols), 0, 0);
p = MakeHuffmanHeader(p, lum_ac_codelens,
    sizeof(lum_ac_codelens),
    lum_ac_symbols,
    sizeof(lum_ac_symbols), 0, 1);
p = MakeHuffmanHeader(p, chm_dc_codelens,
    sizeof(chm_dc_codelens),
    chm_dc_symbols,
    sizeof(chm_dc_symbols), 1, 0);
p = MakeHuffmanHeader(p, chm_ac_codelens,
    sizeof(chm_ac_codelens),
    chm_ac_symbols,
    sizeof(chm_ac_symbols), 1, 1);

*p++ = 0xff;
*p++ = 0xda;      /* SOS */
*p++ = 0;         /* bit de poids fort de la longueur */
*p++ = 12;        /* bit de moindre poids de la longueur */
*p++ = 3;         /* 3 composants */
*p++ = 0;         /* composant 0 */
*p++ = 0;         /* tableau huffman 0 */
*p++ = 1;         /* composant 1 */
*p++ = 0x11;     /* tableau huffman 1 */
*p++ = 2;         /* composant 2 */
*p++ = 0x11;     /* tableau huffman 1 */
*p++ = 0;         /* premier coefficient DCT */
*p++ = 63;        /* dernier coefficient DCT */
*p++ = 0;         /* approximation successive */
return (p - start);
};

```

Appendice C

Le sous programme suivant est utilisé pour illustrer la fragmentation de paquet RTP/JPEG et la création d'en-tête.

Pour être clair et bref, les définitions de structure sont seulement valides pour les architectures de 32 bits grosses boutiennes (octet de poids fort en premier). Les champs sont supposés être empacotés étroitement en ordre gros boutien, sans bourrage supplémentaire. Des modifications vont être requises pour construire une mise en œuvre portable.

```

/*
 * En-tête de données RTP de la RFC1889
 */
typedef struct {

```

```

unsigned int version:2;      /* version du protocole */
unsigned int p:1;           /* fanion de bourrage */
unsigned int x:1;           /* fanion d'extension d'en-tête */
unsigned int cc:4;          /* compte de CSRC */
unsigned int m:1;           /* bit Marqueur */
unsigned int pt:7;          /* type de charge utile */
u_int16 seq;                /* numéro de séquence */
u_int32 ts;                 /* horodatage */
u_int32 ssrc;               /* source de synchronisation */
u_int32 csrc[1];           /* liste facultative de CSRC */
} rtp_hdr_t;

```

```
#définir le RTP_HDR_SZ 12
```

```
/* La définition suivante est de la RFC1890 */
```

```
#définir RTP_PT_JPEG      26
```

```

struct jpeghdr {
    unsigned int tspec:8;      /* champ spécifique du type */
    unsigned int off:24;      /* décalage d'octets de fragment */
    u_int8 type;               /* identifiant des paramètres de décodeur jpeg */
    u_int8 q;                  /* facteur de quantification (ou identifiant de tableau) */
    u_int8 width;              /* largeur de trame en blocs de 8 pixels */
    u_int8 heigth;            /* hauteur de trame en blocs de 8 pixels */
};

```

```

struct jpeghdr_rst {
    u_int16 dri;
    unsigned int f:1;
    unsigned int l:1;
    unsigned int count:14;
};

```

```

struct jpeghdr_qtable {
    u_int8 mbz;
    u_int8 precision;
    u_int16 length;
};

```

```
#définir RTP_JPEG_RESTART    0x40
```

```
/* Procédure SendFrame :
```

```
*
```

```
* Arguments:
```

```
* start_seq : numéro de séquence pour le premier paquet de la trame en cours.
```

```
* ts : horodatage RTP pour la trame en cours
```

```
* ssrc : valeur de SSRC RTP
```

```
* jpeg_data : données d'examen JPEG codées en Huffman
```

```
* len : longueur des données d'examen JPEG
```

```
* type : valeur à laquelle le champ TypRTP/JPEG devrait être réglée
```

```
* typespec : valeur à laquelle le champ spécifique du type RTP/JPEG devrait être réglé
```

```
* width : largeur en pixels de l'image JPEG
```

```
* heigth : hauteur en pixels de l'image JPEG
```

```
* dri : nombre de MCU entre marqueurs de redémarrage (ou 0 si il n'y a pas de marqueur de redémarrage dans les données
```

```
* q : facteur Q des données, à spécifier en utilisant l'algorithme du groupe JPEG indépendant si  $1 \leq q \leq 99$ , spécifié explicitement avec lqt et cqt si  $q \leq 128$ , ou indéfini autrement.
```

```
* lqt : tableau de quantification pour le canal de luminance si  $q \geq 128$ 
```

```
* cqt : tableau de quantification pour les canaux de chrominance si  $q \geq 128$ 
```

```
** Valeur de retour : numéro de séquence à envoyer pour le premier paquet de la prochaine trame.
```

```
*
```

```
* Ce qui suit est supposé être défini :
```

```

*
* TAILLE DE PAQUET - taille du paquet sortant
* send_packet(u_int8 *data, int len) – envoie le paquet au réseau
*/

u_int16 SendFrame(u_int16 start_seq, u_int32 ts, u_int32 ssrc,
                 u_int8 *jpeg_data, int len, u_int8 type,
                 u_int8 typespec, int width, int height, int dri,
                 u_int8 q, u_int8 *lqt, u_int8 *cqt) {
    rtp_hdr_t rtp_hdr;
    struct jpeg_hdr jpeg_hdr;
    struct jpeg_hdr_rst rst_hdr;
    struct jpeg_hdr_qtable qtbl_hdr;
    u_int8 paquet_buf[TAILLE DE PAQUET];
    u_int8 *ptr;
    int octets_left = len;
    int seq = start_seq;
    int pkt_len, data_len;

    /* Initialise l'en-tête RTP */

    rtp_hdr.version = 2;
    rtp_hdr.p = 0;
    rtp_hdr.x = 0;
    rtp_hdr.cc = 0;
    rtp_hdr.m = 0;
    rtp_hdr.pt = RTP_PT_JPEG;
    rtp_hdr.seq = start_seq;
    rtp_hdr.ts = ts;
    rtp_hdr.ssrc = ssrc;

    /* Initialise l'en-tête JPEG */

    jpeg_hdr.tspec = typespec;
    jpeg_hdr.off = 0;
    jpeg_hdr.type = type | ((dri != 0) ? RTP_JPEG_RESTART : 0);
    jpeg_hdr.q = q;
    jpeg_hdr.width = largeur / 8;
    jpeg_hdr.height = hauteur / 8;

    /* Initialise l'en-tête DRI */

    si (dri != 0) {
        rst_hdr.dri = dri;
        rst_hdr.f = 1;           /* ce code n'aligne pas le RI */
        rst_hdr.l = 1;
        rst_hdr.count = 0x3fff;
    }

    /* Initialise l'en-tête de tableau de quantification */

    si (q >= 128) {
        qtbl_hdr.mbz = 0;
        qtbl_hdr.precision = 0; /* ce code utilise seulement des tableaux de 8 bits */
        qtbl_hdr.length = 128; /* 2 tableaux de 64 octets */
    }

    alors que (octets_restants > 0) {
        ptr = paquet_buf + RTP_HDR_SZ;
        memcpy(ptr, &jpeg_hdr, sizeof(jpeg_hdr));
    }
}

```

```

ptr += sizeof(jpghdr);

si (dri != 0) {
    memcpy(ptr, &rsthdr, sizeof(rsthdr));
    ptr += sizeof(rsthdr);
}

si (q >= 128 && jpghdr.off == 0) {
    memcpy(ptr, &qtblhdr, sizeof(qtblhdr));
    ptr += sizeof(qtblhdr);
    memcpy(ptr, lqt, 64);
    ptr += 64;
    memcpy(ptr, cqf, 64);
    ptr += 64;
}

data_len = TAILLE DE PAQUET - (ptr - paquet_buf);
si (data_len >= octets_restants) {
    data_len = octets_restants;
    rtp_hdr.m = 1;
}

memcpy(paquet_buf, &rtp_hdr, RTP_HDR_SZ);
memcpy(ptr, jpeg_data + jpghdr.off, data_len);

send_paquet(paquet_buf, (ptr - paquet_buf) + data_len);

jpghdr.off += data_len;
octets_left -= data_len;
rtp_hdr.seq++;
}
retourne rtp_hdr.seq;
}

```

Appendice D

Cette section résume les changements entre le présent document et son prédécesseur, la RFC 2035. Les changements au protocole ont été faits en essayant de causer aussi peu que possible de problèmes d'interopérabilité entre les mises en œuvre fondées sur l'ancien texte et les mises en œuvre plus récentes, et bien sûr, beaucoup des conventions obsolètes peuvent encore être décodées de façon non ambiguë par une mise en œuvre plus récente. Cependant, l'utilisation des anciennes conventions dans les mises en œuvre récentes est fortement déconseillée.

- o Les types 0 et 1 ont été augmentés pour permettre le codage des images vidéo entrelacées, en utilisant 2 bits du champ spécifique du type. Voir les détails au paragraphe 4.1.
- o Il y a eu une discussion dans le groupe de travail pour avoir plus de souplesse pour spécifier les tableaux de quantification JPEG. Le présent mémoire permet que les coefficients de tableaux soient spécifiés explicitement par l'utilisation d'un en-tête Tableau de quantification facultatif, discuté aux paragraphes 3.1.8 et 4.2.
- o Dans la RFC 2035, le codage des informations de marqueur de redémarrage dans le champ Type rend difficile d'ajouter de nouveaux types. De plus, le champ spécifique de type était utilisé pour le compte de redémarrage, le rendant indisponible pour d'autres objets spécifiques du type. Le présent mémoire déplace l'indication de marqueur de redémarrage dans un bit particulier du champ Type, et ajoute un en-tête facultatif pour contenir les informations supplémentaires requises, laissant le champ spécifique de type libre pour cet objet. Le traitement du décodage de trame partielle a aussi été rendu plus robuste à la perte de paquet. Voir les détails aux paragraphes 3.1.7 et 4.4.

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1998). Tous droits réservés.

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

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

Le présent document et les informations 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.