Groupe de travail Réseau
Request for Comments: 5651
RFC rendue obsolète : 3451
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

M. Luby, Qualcomm, Inc.
M. Watson, Qualcomm, Inc.
L. Vicisano, Qualcomm, Inc.

octobre 2009

Bloc de construction du transport de codage en couches (LCT)

Résumé
Le bloc de construction du transport de codage en couches (LCT, *Layered Coding Transport*) fournit une prise en charge de niveau transport pour la livraison fiable de contenu et les protocoles de livraison de flux. LCT est spécifiquement conçu pour prendre en charge les protocoles qui utilisent la diffusion groupée IP, mais il fournit aussi la prise en charge de protocoles qui utilisent l'envoi individuel. LCT est compatible avec le contrôle d'encombrement qui fournit plusieurs taux de livraison aux receveurs et est aussi compatible avec les techniques de codage qui fournissent une livraison fiable du contenu. Le présent document rend obsolète la RFC 3451.

Statut du présent mémoire
Le présent document spécifie un protocole de l'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 "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.

Table des matières

1. Introduction

Le transport de codage en couches (LCT, *Layered Coding Transport*) fournit la prise en charge au niveau du transport pour les protocoles de livraison fiable de contenu et de livraison de flux. Le transport de codage en couches est spécifiquement conçu pour prendre en charge les protocoles qui utilisent la diffusion groupée IP, mais il fournit aussi la prise en charge de protocoles qui utilisent l'envoi individuel. Le transport de codage en couches est compatible avec le contrôle d'encombrement qui fournit plusieurs taux de livraison aux receveurs et est aussi compatible avec les techniques de codage qui fournissent une livraison fiable du contenu.

Le présent document décrit un bloc de construction comme défini dans [RFC3048]. Le présent document est produit par le groupe de travail RMT de l'IETF et suit les lignes directrices générales de la [RFC3269].

La [RFC3451], qui a été publiée dans la catégorie "Expérimentale" et qui est rendue obsolète par le présent document, contenait une version précédente du protocole.

La présente spécification se fonde donc sur, et est rétro compatible avec, le protocole défini dans la [RFC3451] mis à jour en fonction de l'expérience accumulée et de la maturité croissante du protocole depuis sa première publication. Cette expérience s'applique à cette spécification elle-même et aux stratégies de contrôle d'encombrement relatives à l'utilisation de cette spécification.

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].

2. Raisons

LCT fournit la prise en charge du niveau transport pour les protocoles adaptables qui utilisent le service réseau de diffusion groupée IP. La prise en charge que LCT fournit est commune à diverses applications très importantes, incluant des applications de livraison fiable de contenu et de flux.

Une session LCT comporte plusieurs canaux dont l'origine est un seul envoyeur, qui sont utilisés pendant un certain temps pour porter des paquets relevant de la transmission d'un ou plusieurs objets qui peut intéresser les receveurs. La logique derrière la définition d'une session comme ayant pour origine un seul envoyeur est que c'est la bonne granularité pour réguler le trafic de paquets via le contrôle d'encombrement. Une raison de l'utilisation de plusieurs canaux au sein de la même session est qu'ils sont adaptables aux protocoles de contrôle d'encombrement qui utilisent plusieurs canaux par session. Ces protocoles de contrôle d'encombrement sont considérés comme étant en couches parce que un receveur se joint et quitte les canaux dans un ordre mis en couches durant sa participation à la session.

L'utilisation de canaux en couches est aussi utile pour les applications de flux.

Il y a des techniques de codage qui fournissent une fiabilité adaptable et une livraison asynchrone compatibles avec le contrôle d'encombrement en couches et avec LCT. Quand toutes sont combinées, le résultat est un protocole de livraison de contenu asynchrone fiable et adaptable qui est favorable au réseau. LCT fournit aussi une fonction qui peut être utilisée aussi pour d'autres applications, par exemple, des applications de flux en couches.

LCT évite de fournir une fonction qui ne soit pas adaptable en masse. Par exemple, LCT ne fournit aucun mécanisme pour

envoyer des informations des receveurs aux envoyeurs, bien que cela n'exclue pas les protocoles qui utilisent LCT et exigent aussi d'envoyer des informations des receveurs aux envoyeurs.

LCT inclut une prise en charge générale du contrôle d'encombrement qui doit être utilisé. Il ne spécifie cependant pas quel contrôle d'encombrement devrait être utilisé. La raison en est que le contrôle d'encombrement doit être fourni par tout protocole qui est favorable au réseau, et donc les différentes applications qui peuvent utiliser LCT ne vont pas avoir les mêmes exigences pour le contrôle d'encombrement. Par exemple, un protocole de livraison de contenu peut s'efforcer d'utiliser toute la bande passante disponible entre les receveurs et l'envoyeur. Il doit donc réduire drastiquement son taux d'envoi quand il y a du trafic en compétition. Par ailleurs, un protocole de livraison de flux peut s'efforcer de maintenir un taux constant au lieu d'essayer d'utiliser toute la bande passante disponible, et il peut ne pas réduire son taux aussi vite quand il y a du trafic en compétition.

Au delà de la prise en charge du contrôle d'encombrement, LCT fournit un certain nombre de champs et prend en charge des fonctions couramment exigées par de nombreux protocoles. Par exemple, LCT fournit un identifiant de session de transmission qui peut être utilisé pour identifier à quelle session appartient chaque paquet reçu. Ceci est important parce que un receveur peut être joint sur de nombreuses sessions concurrentes, et donc il est très utile d'être capable de démultiplexer les paquets lorsque ils arrivent en accord avec la session à laquelle ils appartiennent. Autre exemple, il y a des champs facultatifs au sein de l'en-tête de paquet LCT pour identifier l'objet sur lequel des informations sont portées dans la charge utile de paquet.

3. Fonctionnalité

Une session LCT consiste en un ensemble de canaux LCT logiquement groupés associés à un seul envoyeur portant des paquets avec des en-têtes LCT pour un ou plusieurs objets. Un canal LCT est défini par la combinaison d'un envoyeur et d'une adresse associée au canal par l'envoyeur. Un receveur se joint à un canal pour commencer à recevoir les paquets de données envoyés au canal par l'envoyeur, et un receveur quitte un canal pour cesser de recevoir les paquets de données provenant du canal.

LCT est destiné à être combiné avec d'autres blocs de construction afin que le protocole global résultant soit massivement adaptable. L'adaptabilité se réfère au comportement du protocole en relation avec le nombre de receveurs et les chemins de réseau, leur hétérogénéité, et la capacité de s'accommoder d'ensembles de receveurs variant dynamiquement. Des limitations de l'adaptabilité peuvent venir d'exigences de mémoire ou de traitement, ou de la quantité de trafic de paquets de contrôle de rétroaction et de données redondantes générées par le protocole. À leur tour, de telles limitations peuvent être une conséquence des caractéristiques qu'un protocole de livraison de contenu complètement fiable ou de livraison de flux est supposé fournir.

L'en-tête LCT fournit un certain nombre de champs qui sont utiles pour porter des informations de session dans la bande aux receveurs. Un des champs exigés est l'identifiant de session de transmission (TSI, *Transmission Session ID*) qui permet au receveur d'une session d'identifier de façon univoque les paquets reçus au titre de la session. Un autre champ exigé est Informations de contrôle d'encombrement (CCI, *Congestion Control Information*) qui permet au receveur d'effectuer le contrôle d'encombrement exigé sur les paquets reçus au sein de la session. D'autres champs LCT fournissent des informations supplémentaires facultatives mais souvent très utiles pour la session. Par exemple, l'identifiant d'objet de transport (TOI, *Transport Object Identifier*) identifie pour quel objet le paquet contient des données et des fanions sont inclus pour indiquer la fermeture de la session et la fin de l'envoi de paquets pour un objet. Des extensions d'en-tête peuvent porter des champs supplémentaires qui par exemple, peuvent être utilisés pour l'authentification de paquet ou pour porter diverses sortes d'informations relatives au temps : l'heure courante de l'envoyeur (SCT, *Sender Current Time*) porte l'heure à laquelle le paquet a été envoyé de l'envoyeur au receveur, le temps résiduel attendu (ERT, *Expected Residual Time*) porte la durée pendant laquelle la session ou l'objet de transmission va être poursuivie, et le dernier changement de la session (SLC, *Session Last Change*) porte l'heure à laquelle des objets ont été ajoutés, modifiés, ou supprimés de la session.

LCT fournit la prise en charge du contrôle d'encombrement. Le contrôle d'encombrement qui DOIT être utilisé doit se conformer à la [RFC2357] entre receveurs et l'envoyeur pour chaque session LCT. Contrôle d'encombrement se réfère à la capacité d'adapter le débit à la bande passante disponible sur le chemin de l'envoyeur à un receveur, et de partager équitablement la bande passante avec les flux en concurrence, tels que TCP. Donc, le flux total de paquets s'écoulant vers un receveur participant à une session LCT NE DOIT PAS faire une concurrence déloyale aux protocoles à adaptation de flux existants comme TCP.

Un protocole de contrôle d'encombrement à plusieurs débits ou à un seul débit peut être utilisé avec LCT. Pour des protocoles à plusieurs débits, une session consiste normalement en plus d'un canal, et l'envoyeur envoie des paquets aux

canaux dans la session à des débits qui ne dépendent pas des receveurs. Chaque receveur ajuste son taux de réception durant sa participation à la session en se joignant et en quittant dynamiquement les canaux en dépendant de la bande passante disponible pour l'envoyeur indépendamment de tous les autres receveurs. Donc, pour les protocoles à plusieurs débits, le taux de réception de chaque receveur peut varier dynamiquement indépendamment des autres receveurs.

Pour les protocoles à un seul débit, une session consiste normalement en un canal et l'envoyeur envoie les paquets au canal à des débits variables dans le temps selon les retours des receveurs. Chaque receveur reste joint au canal durant sa participation à la session. Donc, pour les protocoles à un seul débit, le taux de réception de chaque receveur peut varier dynamiquement mais en coordination avec tous les receveurs.

Généralement, un protocole à plusieurs débits est préférable à un protocole à un seul débit dans un environnement de receveurs hétérogène, car généralement il réalise plus facilement l'adaptabilité à de nombreux receveurs et fournit un débit plus élevé à chaque receveur individuel. L'utilisation d'un schéma de contrôle d'encombrement à plusieurs débits défini dans la [RFC3738] est RECOMMANDÉ. D'autres protocoles de contrôle d'encombrement à plusieurs débits sont décrits dans [VIC1998] et [BYE2000]. Un protocole de contrôle d'encombrement possible à un seul débit est décrit dans [RIZ2000].

Codage en couches se réfère à la capacité de produire un flux codé de paquets qui peut être partagé en un ensemble ordonné de couches. Le codage est destiné à fournir une certaine forme de fiabilité, et la mise en couches est destinée à permettre au ressenti du receveur (en termes de qualité d'exécution, ou de vitesse globale de transfert) de varier de façon prévisible selon le nombre de couches consécutives de paquets que reçoit le receveur.

Le concept de codage en couches a été introduit en référence aux flux audios et vidéos. Par exemple, les informations associées à une diffusion de TV pourraient être partagées en trois couches, correspondant à la qualité noir et blanc, couleur, et haute définition (HDTV). Les receveurs peuvent obtenir des qualités différentes sans qu'il soit besoin que l'envoyeur duplique les informations dans les différentes couches.

Le concept de codage en couches peut être naturellement étendu aux protocoles de livraison fiable de contenu quand les techniques de correction d'erreur directe (FEC, *Forward Error Correction*) sont utilisées pour coder le flux de données. Les descriptions de cela se trouvent dans [RIZ1997a], [RIZ1997b], [GEM2000], [VIC1998], et [BYE1998]. En utilisant la FEC, le flux des données est transformé d'une façon telle que la reconstruction d'un objet de données ne dépende pas de la réception de paquets de données spécifiques, mais seulement du nombre de différents paquets reçus. Par suite, en augmentant le nombre de couches d'où un receveur reçoit, le receveur peut réduire en conséquence le temps de transfert. Utiliser la FEC pour assurer la fiabilité peut augmenter considérablement l'adaptabilité par rapport à d'autres méthodes pour assurer la fiabilité. Plus de détails sur l'utilisation de FEC pour la livraison fiable de contenu se trouvent dans la [RFC3453].

Les protocoles fiables visent à donner des garanties sur la livraison fiable de données de l'envoyeur aux receveurs prévus. Les garanties varient de la simple intégrité des données de paquet à la livraison fiable d'une copie précise d'un objet à tous les receveurs prévus. Plusieurs protocoles de livraison fiable de contenu ont été construits par dessus la diffusion groupée IP en utilisant des méthodes autres que la FEC, mais l'adaptabilité n'était pas le but principal de la conception de beaucoup d'entre eux.

Deux des difficultés clés de l'adaptation de la livraison fiable de contenu en utilisant la diffusion groupée IP sont de traiter la quantité de données qui s'écoule des receveurs en retour à l'envoyeur et des réponses associées (généralement des retransmissions de données) de la part de l'envoyeur. Les protocoles qui évitent de tels retours, et minimisent la quantité de retransmissions, peuvent être massivement adaptables. LCT peut être utilisé en conjonction avec les codes de FEC ou avec un codec mis en couches pour réaliser la fiabilité avec peu ou pas de retours.

Les instances de protocole (PI, *Protocol Instantiation*) PEUVENT être construites en combinant le cadre LCT avec d'autres composants. Une instance de protocole complète qui utilise LCT DOIT inclure un protocole de contrôle d'encombrement compatible avec LCT et qui se conforme à la [RFC2357]. Une instance de protocole complète qui utilise LCT PEUT inclure un protocole de fiabilité adaptable compatible avec LCT, il PEUT inclure un protocole de contrôle de session compatible avec LCT, et il PEUT inclure d'autres protocoles comme des protocoles de sécurité.

4. Applicabilité

Une session LCT comporte un ensemble en relation logique d'un ou plusieurs canaux LCT dont l'origine est un seul envoyeur. Les canaux sont utilisés pendant un certain temps pour porter des paquets contenant des en-têtes LCT, et ces en-

têtes relèvent de la transmission d'un ou plusieurs objets qui peuvent intéresser les receveurs.

LCT est surtout applicable à la livraison d'objets ou flux dans une session de longueur substantielle, c'est-à-dire, des objets ou flux qui vont d'une gamme de longueur agrégée de centaines de kilo octets à de nombreux giga octets, et où la durée de la session est de l'ordre de la dizaine de secondes ou plus.

Par exemple, une session LCT pourrait être utilisée pour livrer un programme de TV utilisant trois canaux LCT. Recevoir des paquets provenant du premier canal LCT pourrait permettre la réception en noir et blanc. Recevoir les deux premiers canaux LCT pourrait aussi permettre la réception de la couleur. Recevoir tous les trois canaux pourrait permettre la réception de la qualité HDTV. Les objets de cet exemple pourrait correspondre aux programmes individuels de TV transmis.

Comme autre exemple, une session LCT fiable pourrait être utilisée pour livrer fiablement une carte météorologique (objets) mise à jour chaque heure en utilisant dix canaux LCT à des débits différents, en utilisant le codage de FEC. Un receveur peut se joindre et recevoir concurremment les paquets provenant de sous ensembles de ces canaux, jusqu'à ce qu'il ait assez de paquets au total pour récupérer l'objet, puis quitter la session (ou rester connecté en écoutant seulement les informations de description de session) jusqu'au moment de recevoir le prochain objet. Dans ce cas, la métrique de qualité est le temps exigé pour recevoir chaque objet.

Avant de se joindre à une session, les receveurs doivent obtenir assez de la description de session pour commencer la session. Cela inclut les paramètres pertinents de la session nécessaires pour qu'un receveur participe à la session, incluant toutes les informations pertinentes pour le contrôle d'encombrement. La description de session est déterminée par l'envoyeur, et est normalement communiquée hors bande aux receveurs. Dans certains cas, comme décrit plus loin, des parties de la description de session qui ne sont pas exigées pour initier une session PEUVENT être incluses dans l'en-tête LCT ou communiquées hors bande à un receveur après qu'il s'est joint à la session.

Un codeur PEUT être utilisé pour générer les données qui sont placées dans la charge utile de paquet afin d'assurer la fiabilité. Un décodeur convenable est utilisé pour reproduire les informations originales de la charge utile du paquet. Il PEUT y avoir un en-tête de fiabilité qui suit l'en-tête LCT si un tel codeur et décodeur est utilisé. L'en-tête de fiabilité aide à décrire les données de codage portées dans la charge utile du paquet. Le format de l'en-tête de fiabilité dépend du codage utilisé, et ceci est négocié hors bande. Par exemple, un des en-têtes de FEC décrit dans la [RFC5052] pourrait être utilisé.

Pour LCT, quand plusieurs taux de contrôle d'encombrement sont utilisés, le contrôle d'encombrement est réalisé en envoyant les paquets associés à une session donnée à plusieurs canaux LCT. Les receveurs individuels se joignent dynamiquement à un ou plusieurs de ces canaux, selon l'encombrement du réseau perçu par le receveur. Les en-têtes LCT incluent un champ opaque qui DOIT être utilisé pour porter les informations de contrôle d'encombrement aux receveurs. Le schéma réel de contrôle d'encombrement à utiliser avec LCT est négocié hors bande. Des exemples de protocoles de contrôle d'encombrement qui peuvent convenir pour la livraison de contenu sont décrits dans [VIC1998], [BYE2000], et la [RFC3738]. D'autres contrôles d'encombrement peuvent convenir quand LCT est utilisé pour une application de flux.

Le présent document ne spécifie ni ne restreint le type d'échanges entre LCT (ou toute instance de protocole construite par dessus LCT) et une application supérieure. Certaines API supérieures peuvent utiliser une approche en mode objet où la seule unité possible de données échangée entre LCT (ou toute instance de protocole construite par dessus LCT) et une application, à une source ou chez un receveur, est un objet. D'autres API peuvent permettre à une application envoyeuse ou receveuse d'échanger un sous ensemble d'un objet avec LCT (ou toute PI construite par dessus LCT) ou peuvent même suivre un modèle de flux. Ces considérations sortent du domaine d'application du présent document.

4.1 Exigences et considérations environnementales

LCT est destiné à la livraison avec encombrement contrôlé d'objets et flux (à la fois livraison fiable de contenu et flux d'informations multimédia).

LCT peut être utilisé avec la livraison en diffusion groupée et en envoi individuel. LCT exige la connexité entre un envoyeur et les receveurs, mais ne l'exige pas des receveurs à un envoyeur. LCT fonctionne de façon inhérente avec tous les types de réseaux, incluant des LAN, WAN, intranets, l'Internet, réseaux asymétriques, réseaux sans fil, et réseaux par satellite. Donc, l'adaptabilité brute inhérente de LCT n'est pas limitée. Cependant, quand d'autres applications spécifiques sont construites par dessus LCT, ces applications, par leur propre nature, peuvent alors limiter l'adaptabilité. Par exemple, si une application exige que les receveurs restituent des informations hors bande afin de se joindre à une session, ou si une application permet aux receveurs de renvoyer des demandes à l'envoyeur pour rapporter des statistiques de réception, l'adaptabilité de l'application est alors limitée par la capacité d'envoyer, recevoir, et traiter ces données supplémentaires.

LCT exige que les receveurs soient capables d'identifier de façon univoque et démultiplexe les paquets associés à une session LCT. En particulier, il DOIT y avoir un identifiant de session de transport (TSI) associé à chaque session LCT. La portée du TSI est déterminée par l'adresse IP de l'envoyeur, et l'adresse IP de l'envoyeur avec le TSI DOIVENT identifier la session de façon univoque. Si le transport sous-jacent est UDP, comme décrit dans la [RFC0768], alors le numéro d'accès de source UDP de 16 bits PEUT servir de TSI pour la session. La valeur du TSI DOIT être la même dans tous les endroits où il se produit au sein d'un paquet. Si il n'y a pas de TSI sous-jacent fourni par la couche réseau, transport, ou autre, le TSI DOIT alors être inclus dans l'en-tête LCT.

LCT est présumé être utilisé avec un service réseau ou transport sous-jacent qui est un service "au mieux" qui ne garantit pas la réception de paquet ou la réception de paquet dans l'ordre, et qui n'a aucune prise en charge du contrôle de flux ou d'encombrement. Par exemple, le modèle de diffusion groupée toutes sources (ASM, *Any-Source Multicast*) de diffusion groupée IP comme défini dans la [RFC1112] est un tel service réseau "au mieux". Alors que le service de base fourni par la [RFC1112] est largement adaptable, la fourniture du contrôle d'encombrement ou de la fiabilité devrait être faite avec soin pour éviter de sévères limitations d'adaptabilité, en particulier en présence d'ensembles hétérogènes de receveurs.

Il y a actuellement deux modèles de livraison en diffusion groupée, le modèle de diffusion groupée toutes sources (ASM) défini dans la [RFC1112] et le modèle de diffusion groupée spécifique de source (SSM, *Source-Specific Multicast*) comme défini dans la [RFC4607]. LCT fonctionne avec les deux modèles de diffusion groupée, mais d'une façon légèrement différente avec des soucis environnementaux un peu différents. Quand on utilise ASM, un envoyeur S envoie des paquets à un groupe de diffusion groupée G, et l'adresse du canal LCT consiste en la paire (S,G) où S est l'adresse IP de l'envoyeur et G est une adresse de groupe de diffusion groupée. Quand on utilise SSM, un envoyeur S envoie les paquets à un canal SSM (S,G) et l'adresse du canal LCT coïncide avec l'adresse du canal SSM.

Un envoyeur peut allouer localement des adresses uniques de canal SSM, et cela rend l'allocation des adresses de canal LCT facile avec SSM. Pour allouer des adresses de canal LCT en utilisant ASM, l'envoyeur doit choisir de façon unique l'adresse de groupe de diffusion groupée ASM sur la portée du groupe, et cela rend l'allocation des adresses de canal LCT plus difficiles avec ASM.

Les canaux LCT et les canaux SSM coïncident, et donc le receveur va seulement recevoir des paquets envoyés au canal LCT demandé. Avec ASM, le receveur se joint à un canal LCT en se joignant à un groupe de diffusion groupée G, et tous les paquets envoyés à G, sans considération de l'envoyeur, peuvent être reçus par le receveur. Donc, SSM a des avantages de sécurité marquants sur ASM pour la prévention des attaques de déni de service (DoS). Dans l'un et l'autre cas, les receveurs DEVRAIENT utiliser les mécanismes d'authentification de paquet pour atténuer de telles attaques (voir le paragraphe 6.2 et la Section 7).

Certains réseaux ne sont pas accessibles à certains protocoles de contrôle d'encombrement qui pourraient être utilisés avec LCT. En particulier, pour un réseau par satellite ou sans fil, il peut ne pas y avoir de mécanisme pour que les receveurs réduisent effectivement leur taux de réception car il peut y avoir un taux de transmission fixe alloué à la session.

LCT est compatible avec IPv4 et IPv6 car aucune partie du paquet n'est spécifique de la version IP.

4.2 Modèles de service de livraison

LCT peut prendre en charge plusieurs modèles différents de service de livraison. Deux exemples sont brièvement décrits ici.

Modèle de service poussé : une façon d'utiliser un modèle de service poussé peut être la livraison fiable de contenu et la livraison d'une série d'objets. Par exemple, un receveur pourrait se joindre à la session et adapter dynamiquement le nombre de canaux LCT auquel le receveur est joint jusqu'à ce qu'assez de paquets aient été reçus pour reconstruire un objet. Après la reconstruction de l'objet, le receveur peut rester dans la session et attendre la transmission du prochain objet.

Le modèle poussé est particulièrement intéressant dans les réseaux par satellite et sans fil. Dans ces cas, une session peut consister en un canal LCT à débit fixe.

Un modèle de service poussé peut être utilisé, par exemple, pour la livraison fiable d'un gros objet comme un fichier de 100 GB. L'envoyeur pourrait envoyer une annonce de description de session à un canal de contrôle et les receveurs pourraient surveiller ce canal et se joindre à une session chaque fois qu'une description de session intéressante arrive. À réception de la description de session, chaque receveur pourrait se joindre à la session pour recevoir les paquets jusqu'à ce que assez de paquets soient arrivés pour reconstruire l'objet, point auquel le receveur pourrait faire rapport à l'envoyeur que

sa réception s'est achevée avec succès. L'envoyeur pourrait décider de continuer d'envoyer des paquets pour l'objet à la session jusqu'à ce que tous les receveurs aient rapporté la réussite de la reconstruction ou jusqu'à ce que d'autres conditions aient été satisfaites.

Il y a plusieurs caractéristiques de codage en couches asynchrone (ALC) qui fournissent la prise en charge du modèle poussé. Par exemple, l'envoyeur peut facultativement inclure un temps résiduel attendu (ERT) dans l'extension d'en-tête de paquet qui indique la durée restante attendue de transmission de paquet pour le seul objet porté dans la session ou pour l'objet identifié par l'identifiant d'objet de transmission (TOI) si il y a plusieurs objets portés dans la session. Cela peut être utilisé par les receveurs pour déterminer si il reste assez de temps dans la session pour réussir à recevoir assez de paquets supplémentaires pour récupérer l'objet. Si, par exemple, il n'y a pas assez de temps, alors d'application de poussée peut avoir des receveurs qui rapportent à l'envoyeur d'étendre la transmission de paquets pour l'objet à un temps suffisant pour permettre aux receveurs d'obtenir assez de paquets pour reconstruire l'objet. L'envoyeur pourrait alors inclure un ERT sur la base du temps étendu de transmission de l'objet dans chaque en-tête de paquet suivant pour l'objet. Comme autres exemples, l'en-tête LCT peut facultativement contenir un fanion Clôture de session qui indique quand l'envoyeur est sur le point de cesser d'envoyer des paquets à la session et un fanion Clôture d'objet qui indique quand l'envoyeur est sur le point de cesser d'envoyer des paquets à la session pour l'objet identifié par le TOI. Cependant, ces fanions ne sont pas un mécanisme complètement fiable et donc le fanion Clôture de session devrait seulement être utilisé comme indication de quand la session est sur le point de fermer, et le fanion Clôture d'objet devrait seulement être utilisé comme indication de quand la transmission de paquets pour l'objet est sur le point de se terminer.

Modèle de livraison de contenu à la demande : pour un modèle de service de livraison de contenu à la demande, les envoyeurs transmettent normalement pendant une certaine durée choisie comme étant assez longue pour permettre à tous les receveurs prévus de se joindre à la session et récupérer l'objet. Par exemple, une mise à jour de logiciel populaire pourrait être transmise en utilisant LCT sur plusieurs jours, même si un receveur peut être capable d'achever le téléchargement en un total d'une heure de temps de connexion, peut-être étalée sur plusieurs intervalles de temps. Dans ce cas, les receveurs se joignent à la session à tout moment quand elle est active. Les receveurs quittent la session quand ils ont reçu assez de paquets pour récupérer l'objet. Les receveurs, par exemple, obtiennent une description de session en contactant un serveur de la Toile.

Dans ce cas, les receveurs se joignent à la session, et adaptent dynamiquement le nombre de canaux LCT auxquels ils souscrivent selon la bande passante disponible. Les receveurs quittent alors la session quand ils ont reçu assez de paquets pour récupérer l'objet.

Par exemple, supposons qu'un objet fasse 50 MB. L'envoyeur pourrait envoyer 1 kB de paquets au premier canal LCT à 50 paquets par seconde, de sorte que les receveurs qui utilisent juste ce canal LCT pourrait achever la réception de l'objet en 1 000 secondes en l'absence de pertes, et seraient capables d'achever la réception même en présence d'une quantité substantielle de pertes avec l'utilisation du codage pour la fiabilité. De plus, l'envoyeur pourrait utiliser un nombre de canaux LCT tel que le débit agrégé de 1 kB de paquets sue tous les canaux LCT soit de 1 000 paquets par seconde, de sorte qu'un receveur pourrait être capable d'achever la réception de l'objet en seulement 50 secondes (en supposant l'absence de pertes et que le mécanisme de contrôle d'encombrement converge immédiatement sur l'utilisation de tous les canaux LCT).

Autres modèles de service : il y a bien d'autres modèles de service de livraison pour lesquels LCT peut être utilisé, qui ne sont pas traités ci-dessus. Comme exemples, un modèle de service de flux en direct ou de flux d'archivage à la demande de contenus. Une description des nombreuses applications potentielles, le modèle de service de livraison approprié, et les mécanismes supplémentaires pour prendre en charge de telles fonctions quand elles sont combinées avec LCT sort du domaine d'application du présent document. Le présent document tente seulement de décrire le minimum des éléments adaptables communs à ces diverses applications en utilisant LCT comme transport de livraison.

4.3 Contrôle d'encombrement

Le protocole de contrôle d'encombrement spécifique à utiliser pour les sessions LCT dépend du type de contenu à livrer. Bien que le comportement général du protocole de contrôle d'encombrement soit de réduire le débit en présence d'encombrement et de l'augmenter graduellement en l'absence d'encombrement, le comportement dynamique réel (par exemple, la réponse à des pertes uniques) peut varier.

Il est RECOMMANDÉ que le mécanisme de contrôle d'encombrement spécifié dans la [RFC3738] soit utilisé. D'autres protocoles possibles de contrôle d'encombrement pour la livraison fiable de contenu en utilisant LCT sont décrits dans [VIC1998] et [BYE2000]. Des modèles différents de service de livraison pourraient exiger des protocoles de contrôle d'encombrement différents.

5. Champs d'en-tête de paquet

Les paquets envoyés à une session LCT DOIVENT inclure un "en-tête LCT". Le format de l'en-tête LCT est décrit ci-dessous.

D'autres blocs de construction PEUVENT décrire certains des mêmes champs que décrits pour l'en-tête LCT. Il est RECOMMANDÉ que les instances de protocole qui utilisent plusieurs blocs de construction incluent des champs partagés au plus une fois dans chaque paquet. Donc, par exemple, si un autre bloc de construction est utilisé avec LCT qui inclut le champ facultatif Temps résiduel attendu, alors ce champ DEVRAIT être porté dans chaque paquet au plus une fois.

La position de l'en-tête LCT au sein d'un paquet DOIT être spécifiée par une instance de protocole qui utilise LCT.

5.1 Format d'en-tête LCT

L'en-tête LCT est de taille variable, qui est spécifiée par un champ Longueur dans le troisième octet de l'en-tête. Dans l'en-tête LCT, tous les champs d'entiers sont portés dans le format "gros boutien" ou "ordre des octets du réseau", qui est l'octet de poids fort en premier. Les bits désignés comme de "bourrage" ou "réservés" (r) DOIVENT être réglés à 0 par les envoyeurs et ignorés par les receveurs dans cette version de la spécification. Sauf mention contraire, les constantes numériques de cette spécification sont en forme décimale (base 10).

Le format de l'en-tête LCT par défaut est décrit à la Figure 1.

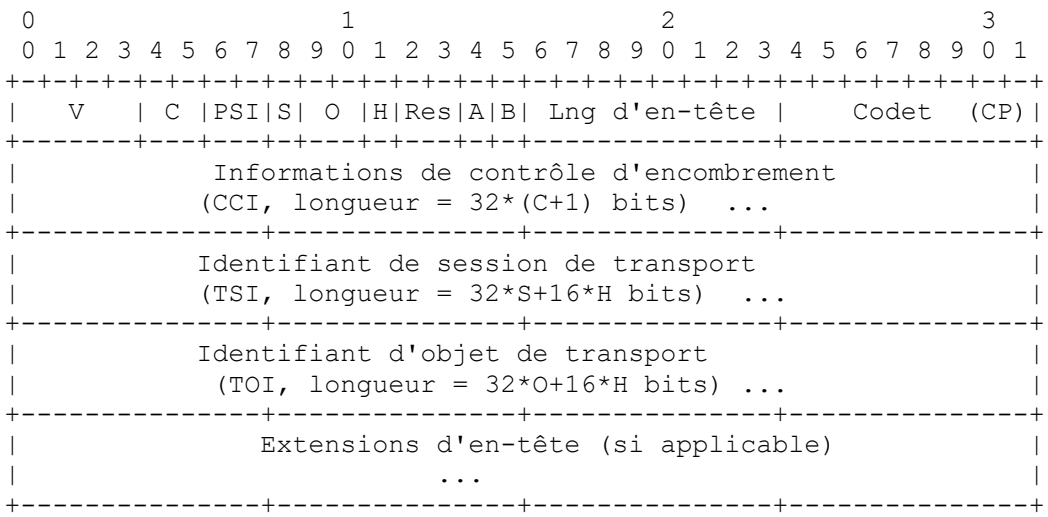

Figure 1 : Format d'en-tête LCT par défaut

La fonction et la longueur de chaque champ de l'en-tête LCT par défaut sont les suivantes :

V, numéro de version LCT : 4 bits. Indique le numéro de version LCT. Le numéro de version LCT pour la présente spécification est 1.

C, fanion contrôle d'encombrement : 2 bits. C=0 indique que le champ Informations de contrôle d'encombrement (CCI) fait 32 bits. C=1 indique que le champ CCI fait 64 bits. C=2 indique que le champ CCI fait 96 bits. C=3 indique que le champ CCI fait 128 bits.

PSI (Indication spécifique du protocole) : 2 bits. L'usage de ces bits, si il en est, est spécifique de chaque instance de protocole qui utilise le bloc de construction LCT. Si aucun usage spécifique d'instance de protocole de ces bits n'est défini, un envoyeur DOIT alors les régler à zéro et un receveur DOIT ignorer ces bits.

S, fanion Identifiant de session de transport : 1 bit. C'est le nombre de mots complets de 32 bits dans le champ TSI. Le champ TSI fait 32*S + 16*H bits, c'est-à-dire, la longueur est 0 bit, 16 bits, 32 bits, ou 48 bits.

O, fanion Identifiant d'objet de transport : 2 bits. C'est le nombre de mots complets de 32 bits dans le champ TOI. Le champ TOI fait 32*O + 16*H bits, c'est-à-dire, la longueur 0 bit, 16 bits, 32 bits, 48 bits, 64 bits, 80 bits, 96 bits, ou 112 bits.

H, fanion Demi mots : 1 bit. Les champs TSI et TOI sont en longueur tous deux des multiples de 32 bits plus 16*H bits. Cela permet que les longueurs des champs TSI et TOI soient des multiples d'un demi mot (16 bits) tout en assurant que la longueur agrégée des champs TSI et TOI champs est un multiple de 32 bits.

Réservé (Res) : 2 bits. Ces bits sont réservés. Dans cette version de la spécification, ils DOIVENT être réglés à zéro par l'envoyeur et être ignorés par les receveurs.

A, fanion Clôture de session : 1 bit. Normalement, A est réglé à 0. L'envoyeur PEUT régler A à 1 quand la fin d'une transmission de paquets pour la session est imminente. A PEUT être réglé à 1 juste dans le dernier paquet transmis pour une session, ou A PEUT être réglé à 1 dans les quelques dernières secondes des paquets transmis pour la session. Une fois que l'envoyeur a réglé A à 1 dans un paquet, l'envoyeur DEVRAIT régler A à 1 dans tous les paquets suivants jusqu'à la fin de la transmission de paquets pour la session. Un paquet reçu avec A réglé à 1 indique à un receveur que l'envoyeur va immédiatement cesser d'envoyer des paquets pour la session. Quand un receveur reçoit un paquet avec A réglé à 1, le receveur DEVRAIT supposer qu'il ne lui sera plus envoyé de paquets pour la session.

B, fanion Clôture d'objet : 1 bit. Normalement, B est réglé à 0. L'envoyeur PEUT régler B à 1 quand la fin de la transmission des paquets pour un objet est imminente. Si le champ TOI est utilisée et si B est réglé à 1, alors la fin de la transmission pour l'objet identifié par le champ TOI est imminente. Si le champ TOI n'est pas utilisé et si B est réglé à 1, alors la terminaison de la transmission pour l'objet de la session identifié par des informations hors bande est imminente. B PEUT être réglé à 1 juste dans le dernier paquet transmis pour l'objet, ou B PEUT être réglé à 1 dans les quelques dernières secondes où les paquets sont transmis pour l'objet. Une fois que l'envoyeur règle B à 1 dans un paquet pour un objet particulier, l'envoyeur DEVRAIT régler B à 1 dans tous les paquets suivants pour l'objet jusqu'à la fin de la transmission des paquets de l'objet.Un paquet reçu avec B réglé à 1 indique à un receveur que l'envoyeur va cesser immédiatement d'envoyer des paquets pour l'objet. Quand un receveur reçoit un paquet avec B réglé à 1, il DEVRAIT alors supposer qu'aucun autre paquet ne va être envoyé pour l'objet sur la session.

Longueur d'en-tête LCT : 8 bits. Longueur totale de l'en-tête LCT en unités de mots de 32 bits. La longueur de l'en-tête LCT DOIT être un multiple de 32 bits. Ce champ peut être utilisé pour accéder directement à la portion du paquet au delà de l'en-tête LCT, c'est-à-dire, au premier autre en-tête si il existe, ou à la charge utile si elle existe et qu'il n'y a pas d'autres en-têtes, ou la fin du paquet si il n'y a pas d'autres en-têtes ou charge utile de paquet.

Codet (CP) : 8 bits. Identifiant opaque qui est passé au décodeur de la charge utile du paquet pour porter les informations sur le codec utilisé pour la charge utile du paquet. La transposition entre le codet et le codec réel est définie session par session et communiquée hors bande au titre des informations de description de session. L'utilisation du champ CP est similaire à celle du champ Type de charge utile (PT, *Payload Type*) dans les en-têtes RTP comme décrit dans la [RFC3550].

Informations de contrôle d'encombrement (CCI) : 32, 64, 96, ou 128 bits. Utilisé pour porter les informations de contrôle d'encombrement. Par exemple, les informations de contrôle d'encombrement pourraient inclure le nombre de couches, le nombre de canaux logiques, et les numéros de séquence. Ce champ est opaque pour les besoins de la présente spécification. Ce champ DOIT faire 32 bits si C=0, 64 bits si C=1, 96 bits si C=2, et 128 bits si C=3.

Identifiant de session de transport (TSI) : 0, 16, 32, ou 48 bits. Le TSI identifie de façon univoque une session parmi toutes les sessions d'un envoyeur particulier. Le TSI a la portée de l'adresse IP de l'envoyeur, et donc l'adresse IP de l'envoyeur et le TSI ensemble identifient de façon univoque la session. Bien qu'un TSI en conjonction avec l'adresse IP de l'envoyeur identifie toujours de façon univoque une session, que le TSI soit ou non inclus dans l'en-tête LCT dépend de ce qui est utilisé comme valeur de TSI. Si le transport sous-jacent est UDP, alors le numéro d'accès de source UDP de 16 bits PEUT servir de TSI pour la session. Si la valeur de TSI apparaît plusieurs fois dans un paquet, toutes les occurrences DOIVENT avoir la même valeur. Si il n'y a pas de TSI sous-jacent fourni par la couche réseau, transport ou autre, le TSI DOIT alors être inclus dans l'en-tête LCT.

Le TSI DOIT être unique parmi toutes les sessions servies par l'envoyeur durant la période où la session est active, et pendant une large période précédant et suivant quand la session est active. Le principal objet du TSI est d'empêcher les receveurs d'accepter par inadvertance des paquets d'un envoyeur qui relève de sessions autres que celles auxquelles les receveurs sont abonnés. Par exemple, supposons qu'une session soit désactivée et qu'ensuite une autre session soit activée par un envoyeur et que les deux sessions utilisent un ensemble de canaux en chevauchement. Un receveur qui se connecte et reste connecté à la première session durant l'activité de cet envoyeur pourrait éventuellement accepter des paquets provenant de la seconde session comme appartenant à la première session si le TSI pour les deux sessions était identique. La transposition des valeurs de champ TSI en sessions sort du domaine d'application du présent document et est à faire hors bande.

La longueur du champ TSI est 32*S + 16*H bits. Noter que la longueur agrégée du champ TSI plus le champ TOI est un multiple de 32 bits.

Identifiant d'objet de transport (TOI) : 0, 16, 32, 48, 64, 80, 96, ou 112 bits. Ce champ indique à quel objet au sein de la session appartient ce paquet. Par exemple, un envoyeur pourrait envoyer un certain nombre de fichiers dans la même session, en utilisant TOI=0 pour le premier fichier, TOI=1 pour le second, etc. Dans un autre exemple, le TOI peut être un identifiant mondial unique de l'objet transmis de plusieurs envoyeurs concurremment, et la valeur de TOI peut être le résultat d'une fonction de hachage appliquée à l'objet. La transposition des valeurs de champ de TOI en objets sort du domaine d'application du présent document et est à faire hors bande. Le champ TOI DOIT être utilisé dans tous les paquets si plus d'un objet est à transmettre dans une session, c'est-à-dire, le champ TOI est soit présent dans tous les paquets d'une session, soit n'est jamais présent. La longueur du champ TOI est 32*O + 16*H bits. Noter que la longueur agrégée du champ TSI plus le champ TOI est un multiple de 32 bits.

5.2 Champs d'extension d'en-tête

5.2.1 Généralités

Les extensions d'en-tête sont utilisées dans LCT pour traiter les champs d'en-tête facultatif qui ne sont pas toujours utilisés ou ont une taille variable. Des exemples d'utilisation d'extensions d'en-tête incluent :

- o des versions de taille étendue de champs d'en-tête déjà existants,
- o des informations d'authentification d'envoyeur et receveur,
- o des informations d'heure de transmission.

La présence des extensions d'en-tête peut être déduite de la longueur de l'en-tête LCT (HDR_LEN). Si HDR_LEN est plus grand que la longueur de l'en-tête standard, alors l'espace d'en-tête restant est pris par des champs d'extension d'en-tête.

Si elles sont présentes, les extensions d'en-tête DOIVENT être traitées pour s'assurer qu'elles sont reconnues avant d'effectuer une procédure de contrôle d'encombrement ou d'accepter autrement un paquet. L'action par défaut pour les extensions d'en-tête non reconnues est de les ignorer. Cela permet la future introduction d'améliorations rétro compatibles à LCT sans changer le numéro de version de LCT. Les extensions d'en-tête non rétro compatibles NE PEUVENT PAS être introduites sans changer le numéro de version de LCT.

Il y a deux formats pour les champs d'extension d'en-tête, comme décrit à la Figure 2. Le premier format est utilisé pour les extensions de longueur variable, avec des valeurs de type d'extension d'en-tête (HET, *Header Extension Type*) entre 0 et 127. Le second format est utilisé pour les extensions de longueur fixe (un mot de 32 bits) en utilisant des valeurs de HET de 127 à 255.

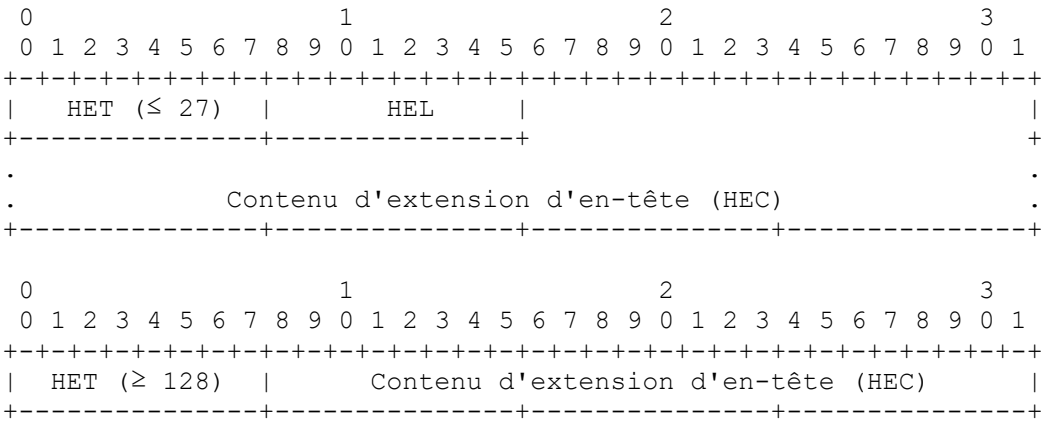

Figure 2 : Format des en-têtes supplémentaires

La signification de chaque sous champ est :

Type d'extension d'en-tête (HET) : 8 bits. Le type de l'extension d'en-tête. Le présent document définit un certain nombre de types possibles. Des types supplémentaires pourront être définis dans de futures versions de cette spécification. Les valeurs de HET de 0 à 127 sont utilisées pour les extension d'en-tête de longueur variable. Les valeurs de HET de 128 à 255 sont utilisées pour les extensions d'en-tête de longueur fixe de 32 bits.

Longueur d'extension d'en-tête (HEL) : 8 bits. Longueur du champ entier d'extension d'en-tête, exprimée en multiples de mots de 32 bits. Ce champ DOIT être présent pour les extensions de longueur variable (HET entre 0 et 127) et NE DOIT PAS être présent pour les extensions de longueur fixe (HET entre 128 et 255).

Contenu d'extension d'en-tête (HEC) : longueur variable. Le contenu de l'extension d'en-tête. Le format de ce sous champ dépend du type d'extension d'en-tête. Pour les extensions d'en-tête de longueur fixe, le HEC est de 24 bits. Pour les extensions d'en-tête de longueur variable, le champ HEC a une taille variable, comme spécifié par le champ HEL. Noter que la longueur de chaque champ d'extension d'en-tête DOIT être un multiple de 32 bits. Noter aussi que la taille totale de l'en-tête LCT, incluant toutes les extensions d'en-tête et tous les champs d'en-tête facultatifs, ne peut pas excéder 255 mots de 32 bits.

Les extensions d'en-tête LCT suivantes sont définies par la présente spécification :

EXT_NOP, HET=0 : extension No-Operation. Les informations présentes dans ce champ d'extension DOIVENT être ignorées par les receveurs.

EXT_AUTH, HET=1 : extension Authentification de paquet. Les informations utilisées pour authentifier l'envoyeur du paquet. Le format de cette extension d'en-tête et son traitement sortent du domaine d'application du présent document et sont à communiquer hors bande au titre de la description de session. Il est RECOMMANDÉ que les envoyeurs fournissent une forme d'authentification de paquet. Si EXT_AUTH est présent, toute vérification d'authentification de paquet qui peut être effectuée immédiatement à réception du paquet DEVRAIT être effectuée avant d'accepter le paquet et d'effectuer sur lui une action relative au contrôle d'encombrement. Certains schémas d'authentification de paquet imposent un délai de plusieurs secondes entre la réception d'un paquet et le moment où le paquet est pleinement authentifié. Toute action relative au contrôle d'encombrement qui est appropriée NE DEVRAIT PAS être retardée par l'authentification de paquet.

EXT_TIME, HET=2 : Extension de temps. Cette extension est utilisée pour porter plusieurs types d'informations de temps. Elle inclut des informations de temps d'objet général, à savoir les extensions de temps Heure courante de l'envoyeur (SCT, *Sender Current Time*), Temps résiduel attendu (ERT, *Expected Residual Time*) et Dernier changement chez l'envoyeur (SLC, *Sender Last Change*) décrites dans le présent document. Elle peut aussi être utilisée pour des informations de temps d'une applicabilité plus étroite (par exemple, définie pour une seule instance de protocole) ; dans ce cas, elles vont être décrites dans un document distinct.

Tous les envoyeurs et receveurs qui mettent en œuvre LCT DOIVENT prendre en charge l'extension d'en-tête EXT_NOP et DOIVENT reconnaître EXT_AUTH et EXT_TIME, mais ne sont pas obligés d'être capables d'analyser leur contenu.

5.2.2 Extension d'en-tête EXT_TIME

Ce paragraphe définit les extensions d'en-tête de temps avec une applicabilité générale. Les valeurs de temps portées dans cette extension d'en-tête sont relatives à l'horloge du serveur. Le serveur DOIT maintenir une heure relative cohérente durant une session (c'est-à-dire, une dérive d'horloge insignifiante). Pour certaines applications, la synchronisation du système ou même mondiale de l'horloge du serveur peut être désirable, comme en utilisant le protocole de l'heure du réseau (NTP, *Network Time Protocol*) [RFC1305] pour assurer l'heure actuelle par rapport à 00:00 GMT, au 1er janvier 1900. Une telle synchronisation externe à la session sort du domaine d'application du présent document.

L'extension d'en-tête EXT_TIME utilise le format décrit à la Figure 3.

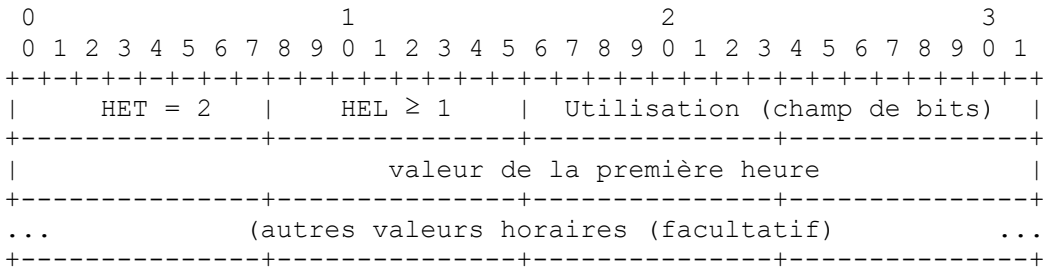

Figure 3: EXT_TIME extension d'en-tête Format

Le champ de bits "Utilisation" indique la signification de la ou des valeurs de temps de 32 bits suivantes.
Il est divisé en deux parties :

- 8 bits sont réservés pour des informations de temps d'objet général. Ces informations sont applicables à tout protocole qui utilise LCT.
- 8 bits sont réservés pour des informations de temps spécifiques de PI. Ces informations sortent du domaine d'application du présent document.

Le format du champ de bits "Utilisation" est décrit à la Figure 4.

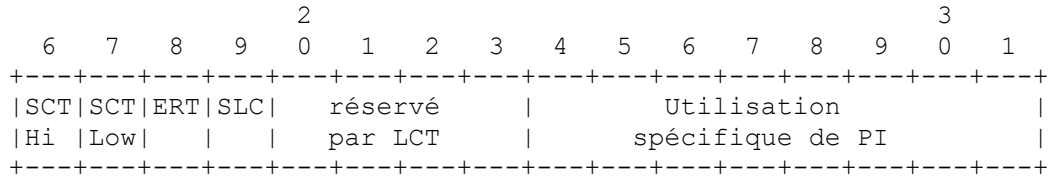

Figure 4 : Format du champ de bits "Utilisation"

Plusieurs champs "valeur de temps" PEUVENT être présents dans une extension d'en-tête EXT_TIME donnée, comme spécifié dans le champ "Utilisation". Quand plusieurs champs "valeur de temps" sont présents, ils DOIVENT apparaître dans l'ordre spécifié par la position de fanion associée dans le champ "Utilisation" : d'abord SCT-High (si présent) puis SCT-Low (si présent) puis ERT (si présent) puis SLC (si présent). Les receveurs DEVRAIENT ignorer les champs supplémentaires au sein de l'extension d'en-tête EXT_TIME qu'ils ne prennent pas en charge.

Les champs pour les informations de temps EXT_TIME d'objet général sont :

Heure courante de l'envoyeur (SCT) : fanion SCT-High, fanion SCT-Low, valeur de temps correspondante (un ou deux mots de 32 bits). Ces informations de temps représentent l'heure courante chez l'envoyeur au moment de la transmission de ce paquet. Quand le fanion SCT-High est établi, la valeur de temps associée de 32 bits donne un entier non signé représentant l'heure en secondes de l'horloge de l'envoyeur. Dans le cas particulier où NTP est utilisé, ces 32 bits donnent un entier non signé représentant le temps en secondes par rapport à 00:00 GMT, le 1er janvier 1900, (c'est-à-dire, les 32 bits de poids fort d'une valeur horaire complète de 64 bits de NTP). Dans ce cas, le traitement du retour à zéro de l'heure de 32 bits sort du domaine de NTP et LCT. Quand le fanion SCT-Low est établi, la valeur d'heure de 32 bits associée fournit un entier non signé représentant un multiple de 1/2^^32 de seconde, afin de permettre une précision inférieure à la seconde. Quand le fanion SCT-Low est établi, le fanion SCT-High DOIT être aussi établi. Dans le cas particulier où NTP est utilisé, ces 32 bits donnent les 32 bits de moindre poids d'un horodatage NTP de 64 bits.

Temps résiduel attendu (ERT) : le fanion ERT, qui correspond à une valeur de temps de 32 bits. Cette information de temps représente le temps de transmission résiduel attendu de l'envoyeur pour la transmission de l'objet en cours. Si le paquet contenant l'information d'ERT contient aussi le champ TOI, alors ERT se réfère à l'objet correspondant au champ TOI ; autrement, il se réfère au seul objet dans la session. Quand le fanion ERT est établi, il est exprimé comme un nombre de secondes. Les 32 bits donnent un entier non signé représentant ce nombre de secondes.

Dernier changement de la session (SLC) : fanion SLC, qui correspond à une valeur de temps de 32 bits. La valeur de l'heure de SLC est celle de l'horloge du serveur, en secondes, à laquelle le dernier changement aux données de la session s'est produit. C'est-à-dire, il exprime l'heure à laquelle le dernier ajout (le plus récent) modification, ou suppression d'objet de transport a été fait pour la session de livraison. Dans le cas de modifications et d'ajouts, il indique que de nouvelles données vont être transportées qui ne l'étaient pas auparavant. Dans le cas de suppressions, SLC indique que des données antérieures ne vont plus être transportées. Quand le fanion SLC est établi, la valeur de temps de 32 bits associée fournit un entier non signé représentant un temps en secondes. Dans le cas particulier où NTP est utilisé, ces 32 bits donnent un entier non signé qui représente le temps en secondes par rapport à 00:00 GMT, le 1er janvier 1900, (c'est-à-dire, les 32 bits de poids fort d'une valeur de temps NTP complète de 64 bits). Dans ce cas, le traitement du retour à zéro de l'heure de 32 bits sort du domaine de NTP et LCT. Dans certains cas, il peut être approprié qu'un paquet qui contient une extension d'en-tête EXT_TIME avec des informations de SLC contienne aussi les information de SCT-High.

Réservé par LCT pour une utilisation future (4 bits) : dans cette version de la spécification, ces bits DOIVENT être réglés à zéro à l'envoi et DOIVENT être ignorés des receveurs.

Utilisation spécifique de PI (8 bits) : ces bits sortent du domaine d'application de ce document. Les bits qui ne sont pas

spécifiés par le PI construit par dessus LCT DEVRAIENT être réglés à zero.

La longueur totale de EXT_TIME est portée dans le HEL, car cette extension d'en-tête est de longueur variable. Elle permet aussi aux clients de sauter cette extension d'en-tête si elle n'est pas prise en charge (mais reconnue).

6. Fonctionnement

6.1 Fonctionnement de l'envoyeur

Avant de se joindre à une session LCT, un receveur DOIT obtenir une description de session. La description de session DOIT inclure :

- l'adresse IP de l'envoyeur,
- le nombre de canaux LCT,
- les adresses et numéros d'accès utilisés pour chaque canal LCT,
- l'identifiant de session de transport (TSI) à utiliser pour la session,
- assez d'informations pour déterminer le protocole de contrôle d'encombrement utilisé,
- assez d'informations pour déterminer le schéma d'authentification de paquet utilisé (si il en est un).

La description de session pourrait aussi inclure, mais sans s'y limiter :

- les débits de données utilisés pour chaque canal LCT,
- la longueur de la charge utile du paquet,
- la transposition des valeurs de TOI en objets pour la session,
- toutes les informations pertinentes pour chaque objet transporté, comme quand il va être disponible au sein de la session, pour combien de temps, et la longueur de l'objet.

Les instances de protocole qui utilisent LCT PEUVENT mettre des exigences supplémentaires sur ce qui doit être inclus dans la description de session. Par exemple, une instance de protocole pourrait exiger que les débits de données pour chaque canal, ou la transposition des valeurs de TOI en objets pour la session, ou d'autres informations relatives aux autres en-têtes qui pourraient être exigés, soient inclus dans la description de session.

La description de session pourrait être sous forme SDP, comme défini dans la [RFC4566], ou un autre format approprié pour une application particulière. Elle pourrait être portée dans un protocole d'annonce de session comme SAP comme défini dans la [RFC2974], obtenu en utilisant un protocole de contrôle de session propriétaire, situé sur une page de la Toile avec des informations de programmation, ou portée via la messagerie électronique ou autres méthodes hors bande. La discussion du format de description de session, et la distribution des descriptions de session sort du domaine d'application de ce document.

Dans une session LCT, un envoyeur qui utilise LCT transmet une séquence de paquets, chacun du format défini ci-dessus. Les paquets sont envoyés d'un envoyeur qui utilise un ou plusieurs canaux LCT, qui ensemble constituent une session. Les taux de transmission peuvent être différents dans les différents canaux et peuvent varier dans le temps. La spécification des autres en-têtes de bloc de construction et de la charge utile de paquet utilisée par une instance de protocole complète utilisant LCT sort du domaine d'application de ce document. Le présent document ne spécifie pas l'ordre de transmission des paquets, ni l'organisation d'une session en plusieurs canaux. Bien que ces questions affectent l'efficacité du protocole, elles n'affectent pas la correction ni l'inter-opérabilité de LCT entre envoyeurs et receveurs.

Plusieurs objets peuvent être portés au sein de la même session LCT. Dans ce cas, chaque objet DOIT être identifié par un TOI unique. Les objets PEUVENT être transmis à la suite, ou ils PEUVENT être transmis concurremment. Il est de bonne pratique d'envoyer seulement les objets en concurrence dans la même session si les receveurs qui participent à cette portion de la session ont un intérêt à recevoir tous les objets. La raison en est que cela gaspille la bande passante et les ressources du réseau pour que des receveurs reçoivent des données pour des objets qui ne les intéressent pas.

Normalement, le ou les envoyeurs continuent d'envoyer des paquets dans une session jusqu'à ce que la transmission soit considérée comme terminée. La transmission peut être considérée terminée quand un certain temps s'est écoulé, qu'un certain nombre de paquets ont été envoyés, ou qu'un signal hors bande (éventuellement d'un protocole de niveau supérieur) a indiqué l'achèvement par un nombre de receveurs suffisant.

Pour les raisons mentionnées ci-dessus, le présent document ne met aucune restriction aux tailles de paquet. Cependant, des considérations d'efficacité du réseau recommandent que l'envoyeur utilise une taille de charge utile de paquet aussi grande que possible, mais d'une façon telle que les paquets n'excèdent pas la taille de l'unité de transmission maximum (MTU, *maximum transmission unit*) du réseau, ou quand la fragmentation couplée avec des pertes de paquet pourrait introduire une

sévère inefficacité dans la transmission.

Il est recommandé que tous les paquets aient la même taille ou des tailles très similaires, car cela peut avoir un impact sévère sur l'efficacité des schémas de contrôle d'encombrement comme ceux décrits dans [VIC1998], [BYE2000], et la [RFC3738]. Un envoyeur de paquets qui utilise LCT DOIT mettre en œuvre la partie côté envoyeur d'un des schémas de contrôle d'encombrement qui se conforment à la [RFC2357] en utilisant le champ Informations de contrôle d'encombrement fourni dans l'en-tête LCT, et le schéma correspondant de contrôle d'encombrement de receveur est à communiquer hors bande et DOIT être mis en œuvre par tout receveur participant à la session.

6.2 Fonctionnement du receveur

Les receveurs peuvent opérer différemment selon le modèle de service de livraison. Par exemple, pour un modèle de service à la demande, les receveurs peuvent se joindre à une session, obtenir les paquets nécessaires pour reproduire l'objet, et ensuite quitter la session. Autre exemple, pour un modèle de service de flux, un receveur peut se joindre continuement à un ensemble de canaux LCT pour télécharger tous les objets dans une session.

Pour être capable de participer à une session, un receveur DOIT obtenir les informations de description de session pertinentes mentionnées au paragraphe 6.1.

Si les informations d'authentification de paquet sont présentes dans un en-tête LCT, elles DEVRAIENT être utilisées comme spécifié au paragraphe 5.2. Pour être capable d'être un receveur dans une session, le receveur DOIT être capable de traiter l'en-tête LCT. Le receveur DOIT être capable d'éliminer, transmettre, mémoriser, ou traiter les autres en-têtes et la charge utile du paquet. Si un receveur n'est pas capable de traiter un en-tête LCT, il DOIT quitter la session.

Pour être capable de participer à une session, un receveur DOIT mettre en œuvre le protocole de contrôle d'encombrement spécifié dans la description de session en utilisant le champ Informations de contrôle d'encombrement fourni dans l'en-tête LCT. Si un receveur n'est pas capable de mettre en œuvre le protocole de contrôle d'encombrement utilisé dans la session, il NE DOIT PAS se joindre à la session. Quand la session est transmise sur plusieurs canaux LCT, les receveurs DOIVENT initialement se joindre aux canaux en accord avec le comportement de départ spécifié par le protocole de contrôle d'encombrement. Pour un protocole de contrôle d'encombrement à plusieurs débits qui utilise plusieurs canaux, cela veut normalement dire qu'un receveur va initialement se joindre seulement à un ensemble minimal de canaux LCT, éventuellement un seul, qui agrégés portent les paquets à bas débit. Cette règle a pour objet d'empêcher les receveurs de commencer à des débits de données élevés.

Plusieurs objets peuvent être portés à la suite ou concurremment au sein de la même session LCT. Dans ce cas, chaque objet est identifié par un TOI unique. Noter que même si un serveur cesse d'envoyer des paquets pour un vieil objet avant de commencer à transmettre des paquets pour un nouvel objet, le réseau et les couches de protocole sous-jacentes peuvent tous deux causer un nouvel ordre des paquets, en particulier quand ils sont envoyés sur des canaux LCT différents, et donc les receveurs NE DEVRAIENT PAS supposer que la réception d'un paquet pour un nouvel objet signifie qu'il n'y a plus de paquets en transit pour le précédent, au moins pendant un certain temps.

Un receveur PEUT se joindre concurremment à plusieurs sessions LCT provenant d'un ou plusieurs envoyeurs. Le receveur DOIT effectuer le contrôle d'encombrement sur chacune de ces sessions LCT. Si le protocole de contrôle d'encombrement permet au receveur une certaine souplesse en termes d'action au sein d'une session, alors le receveur PEUT faire des choix pour optimiser les performances de flux de paquets à travers plusieurs sessions LCT, tant que le receveur respecte les règles de contrôle d'encombrement pour chaque session LCT individuelle.

7. Exigences provenant d'autres blocs de construction

Comme décrit dans la [RFC3048], LCT est un bloc de construction destiné à être utilisé en conjonction avec d'autres blocs de construction, pour spécifier une instance de protocole. Un bloc de construction de contrôle d'encombrement qui utilise le champ Informations de contrôle d'encombrement au sein de l'en-tête LCT DOIT être utilisé par toute instance de protocole qui utilise LCT ; d'autres blocs de construction PEUVENT aussi être utilisés, comme un bloc de construction de fiabilité.

Le contrôle d'encombrement DOIT être appliqué à la session LCT comme une entité, c'est-à-dire, sur l'agrégat de trafic porté par tous les canaux LCT associés à la session LCT. Le champ Informations de contrôle d'encombrement dans l'en-tête LCT est un champ opaque réservé au transport d'informations relatives au contrôle d'encombrement. Il PEUT aussi y avoir de champs d'en-tête d'extension de contrôle d'encombrement qui portent des informations supplémentaires relatives au

contrôle d'encombrement.

Le codeur en couches particulier et les protocoles de contrôle d'encombrement utilisés avec LCT ont un impact sur les performances et l'applicabilité de LCT. Par exemple, des codeurs en couches utilisés pour des flux vidéo et audio peuvent produire un nombre de couches très limité, fournissant donc un contrôle très grossier du taux de réception des paquets par les receveurs d'une session. Quand LCT est utilisé pour le transfert fiable de données, certains codecs de FEC sont limités par nature dans la taille de l'objet qu'ils peuvent coder, et pour les objets plus grands que cette taille les frais généraux de réception chez les receveurs peuvent augmenter de façon substantielle.

Une description plus en profondeur de l'utilisation de la FEC dans les protocoles de transport fiable de diffusion groupée (RMT, *Reliable Multicast Transport*) est donnée dans la [RFC3453]. Certains des codecs de FEC qui PEUVENT être utilisés en conjonction avec LCT pour la livraison fiable de contenu sont spécifiés dans la [RFC5052]. Le champ Codet dans l'en-tête LCT est un champ opaque qui peut être utilisé pour porter des informations relatives au codage de la charge utile du paquet.

LCT exige aussi des receveurs qu'ils obtiennent une description de session, comme décrit au paragraphe 6.1. La description de session pourrait être de forme SDP comme défini dans la [RFC4566], ou dans un autre format approprié pour une application particulière, et peut être distribuée avec SAP comme défini dans la [RFC2974], en utilisant HTTP, ou d'autres façons. Il est RECOMMANDÉ qu'un protocole d'authentification soit utilisé pour livrer la description de session aux receveurs pour assurer que la description de session correcte arrive.

Il est RECOMMANDÉ que les mises en œuvre de LCT utilisent un schéma d'authentification de paquet pour protéger le protocole des attaques. Un exemple d'un schéma potentiellement convenable est décrit dans [Perrig2001].

Certaines instances de protocole qui utilisent LCT PEUVENT utiliser des blocs de construction qui exigent la génération de retours des receveurs à l'envoyeur. Cependant, le mécanisme pour le faire sort du domaine d'application de LCT.

8. Considérations sur la sécurité

LCT est un bloc de construction comme défini dans la [RFC3048] et à ce titre ne définit pas un protocole complet. Les instances de protocole qui utilisent le bloc de construction LCT DOIVENT traiter les vulnérabilités potentielles décrites dans les paragraphes qui suivent. Pour un exemple, voir la [RFC5775].

Les instances de protocole pourraient traiter les vulnérabilités décrites ci-dessous en prenant des mesures pour empêcher les receveurs d'accepter des paquets incorrects, par exemple, en utilisant un mécanisme d'authentification de source et d'intégrité du contenu. Voir aussi le paragraphe 6.2 et la Section 7 pour une discussion des exigences d'authentification de paquet.

Noter que pour un fonctionnement correct, LCT suppose la disponibilité des informations de description de session (voir les Sections 4 et 7). Des informations de description de session incorrectes, ou modifiées par malveillance, peuvent résulter en ce que les receveurs soient incapables de recevoir correctement le contenu de la session, ou que les receveurs essayent par inadvertance de recevoir à un débit plus élevé que ce dont ils sont capables, perturbant ainsi le trafic dans des portions du réseau. Les instances de protocole DOIVENT traiter cette vulnérabilité potentielle, par exemple, en fournissant des mécanismes d'authentification de source et d'intégrité pour la description de session. De plus, ces mécanismes DOIVENT permettre aux receveurs de vérifier de façon sûre la correspondance entre la description de session et les paquets de données LCT.

Les paragraphes qui suivent examinent plus en détails chaque service fourni par LCT.

8.1 Multiplexage et terminaison de session et d'objet

L'identifiant de session de transport et l'identifiant d'objet de transport dans l'en-tête LCT assurent le multiplexage des sessions et des objets. La modification de ces champs par un attaquant pourrait avoir pour effet de priver de données une session ou un objet et potentiellement de diriger des données incorrectes sur une autre session ou objet, effectuant dans les deux cas une attaque de déni de service.

De plus, l'injection de paquets falsifiés avec de fausses valeurs de TSI ou TOI peut causer l'allocation par les receveurs de ressources pour des sessions ou objets supplémentaires, effectuant là encore une potentielle attaque de DoS.

Les bits Clôture d'objet et Clôture de session dans l'en-tête LCT assurent la signalisation de la fin d'une session ou objet. La modification de ces champs par un attaquant pourrait causer un comportement incorrect des receveurs comme si la session ou l'objet était terminé, résultant en une attaque de déni de service, ou à l'inverse de continuer à utiliser sans nécessité des ressources après la fin de la session ou de l'objet (bien que l'utilisation de ressources dans ce cas soit largement une question de mise en œuvre).

Par suite de ces vulnérabilités, ces champs DOIVENT être protégés par les mécanismes de sécurité de l'instance de protocole (par exemple, des mécanismes d'authentification de source et d'intégrité des données).

8.2 Synchronisation

Les mécanismes SCT et ERT fournissent des caractéristiques rudimentaires de synchronisation qui peuvent être soumis à des attaques. Bien sûr un attaquant peut facilement désynchroniser les clients, envoyant des informations de SCT erronées, ou montant une attaque de DoS en informant tous les clients qu'une session (respectivement, un objet particulier) est sur le point d'être close.

Par suite des vulnérabilités ci-dessus, ces champs DOIVENT être protégés par des mécanismes de sécurité d'instance de protocole (par exemple, des mécanismes d'authentification de source et d'intégrité des données).

8.3 Transport des données

Le protocole LCT fournit le transport d'informations pour d'autres blocs de construction, spécifiquement le champ PSI pour l'instance de protocole, le champ Contrôle d'encombrement pour le bloc de construction Contrôle d'encombrement, le champ Codet pour le bloc de construction FEC, l'extension d'en-tête EXT-AUTH (utilisé par l'instance de protocole) et la charge utile de paquet elle-même.

La modification d'un de ces champs par un attaquant peut résulter en une attaque de déni de service. En particulier, la modification du codet ou de la charge utile de paquet peuvent empêcher la réussite de la reconstruction ou causer une reconstruction inappropriée de larges portions d'un objet par les receveurs. La modification du champ Contrôle d'encombrement peut causer la tentative par les receveurs de recevoir à un débit incorrect, empirant potentiellement ou causant une situation d'encombrement et effectuant par là une attaque de DoS.

Par suite des vulnérabilités ci-dessus, ces champs DOIVENT être protégés par les mécanismes de sécurité d'instance de protocole (par exemple, des mécanismes d'authentification de source et d'intégrité des données).

9. Considérations relatives à l'IANA

9.1 Déclaration d'espace de noms pour les types d'extension d'en-tête LCT

Le présent document définit un nouvel espace de noms pour les "types d'extension d'en-tête LCT". Les valeurs dans cet espace de noms sont des entiers entre 0 et 255 (inclus).

Les valeurs dans la gamme de 0 à 63 (inclus) sont réservées à l'usage des extensions d'en-tête LCT de longueur variable et les allocations devront être faites par "revue de l'IETF" comme défini dans la [RFC5226].

Les valeurs dans la gamme de 64 à 127 (inclus) sont réservées à l'usage des extensions d'en-tête LCT de longueur variable et les allocations devront être faites sur la base de "spécification exigée" comme défini dans la [RFC5226].

Les valeurs dans la gamme de 128 à 191 (inclus) sont réservées à l'usage des extensions d'en-tête LCT de longueur fixe et les allocations devront être faites par "revue de l'IETF" comme défini dans la [RFC5226].

Les valeurs dans la gamme de 192 à 255 (inclus) sont réservées à l'usage des extensions d'en-tête LCT de longueur fixe et les allocations devront être faites sur la base de "spécification exigée" comme défini dans la [RFC5226].

Les valeurs initiales du registre des types d'extension d'en-tête LCT sont définies au paragraphe 9.2.

Noter que la version expérimentale précédente de cette spécification réservait les valeurs dans les gammes [64, 127] et [192, 255] pour les extensions d'en-tête LCT spécifiques de PI. Dans l'intérêt de la simplification et comme il n'y avait pas

de recouvrement d'allocations de ces valeurs de type d'extension d'en-tête LCT par les PI, le présent document spécifie un seul espace plat pour les types d'extension d'en-tête LCT.

9.2 Enregistrement de type d'extension d'en-tête LCT

Le présent document enregistre trois valeurs dans l'espace de noms de type d'extension d'en-tête LCT comme suit :

Valeur	Nom	Référence
0	EXT_NOP	RFC5651
1	EXT_AUTH	RFC5651
2	EXT_TIME	RFC5651

10. Remerciements

La présente spécification est en substance fondée sur la [RFC3451] et donc le crédit d'auteurs du présent document est principalement dû aux auteurs de la RFC 3451 : Mike Luby, Jim Gemmel, Lorenzo Vicisano, Luigi Rizzo, Mark Handley, et Jon Crowcroft. Bruce Lueckenhoff, Hayder Radha, et Justin Chapweske ont aussi contribué à la RFC 3451. Des remerciements supplémentaires sont dus à Vincent Roca, Rod Walsh, et Toni Paila pour leurs contributions à la mise à jour de cette proposition de norme.

11. Changements par rapport à la RFC 3451

Cette section résume les changements faits à la version expérimentale de la présente spécification, publiée comme [RFC3451]

- Suppression de la "déclaration d'intention" de l'introduction. (Elle était destinée à préciser le statut "expérimental" de la RFC 3451.)
- Inclusion de matériel de ALC qui est applicable au contexte plus général de LCT.
- Création d'un registre IANA pour les extensions d'en-tête LCT.
- Allocation de deux bits "réservés" dans l'en-tête LCT comme "Indication spécifique du protocole-" - dont l'usage sera défini par les instances de protocole.
- Suppression des champs d'en-tête LCT Heure actuelle d'envoi (SCT) et Temps résiduel attendu (ERT).
- Inclusion d'un nouvel d'en-tête d'extension, EXT_TIME, pour remplacer SCT et ERT et fournir des capacités de temporisation pour des futures extensions.

12. Références

12.1 Références normatives

[RFC0768] J. Postel, "Protocole de datagramme d'utilisateur (UDP)", (STD 6), 28 août 1980.

[RFC1112] S. Deering, "Extensions d'hôte pour diffusion groupée sur IP", STD 5, août 1989. *(Mise à jour par la* RFC2236*)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. DOI 10.17487/RFC2119, *(MàJ par* RFC8174)

[RFC5052] M. Watson et autres, "Bloc de construction de la correction d'erreur directe (FEC)", août 2007. (*Remplace* RFC3452) *(P.S.)*

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008. (*Remplace* RFC2434 *; remplacée par* RFC8126)

12.2 Références pour information

[BYE1998] Byers, J., Luby, M., Mitzenmacher, M., and A. Rege, "Fountain Approach to Reliable Distribution of Bulk Data", Proceedings ACM SIGCOMM'98, Vancouver, Canada, septembre 1998.

[BYE2000] Byers, J., Frumin, M., Horn, G., Luby, M., Mitzenmacher, M., Rotter, A., and W. Shaver, "FLID-DL: Congestion Control for Layered Multicast", Proceedings of Second International Workshop on Networked Group Communications (NGC 2000), Palo Alto, CA, novembre 2000.

[GEM2000] Gemmell, J., Schooler, E., and J. Gray, "Fcast Multicast File Distribution", IEEE Network, Vol. 14, No. 1, pp. 58-68, janvier 2000.

[Perrig2001] Perrig, A., Canetti, R., Song, D., and J. Tyger, "Efficient and Secure Source Authentication for Multicast", Network and Distributed System Security Symposium, NDSS 2001, pp. 35-46, février 2001.

[RFC1035] P. Mockapetris, "Noms de domaines – Mise en œuvre et spécification", DOI 10.17487/RFC1035, STD 13, novembre 1987. (*MàJ par* RFC1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604, 7766, 8482, 8767)

[RFC2357] A. Mankin, A. Romanov, S. Bradner et V. Paxson, "Critères de l'IETF pour l'évaluation des protocoles de transport et d'application de diffusion groupée fiable", juin 1998. *(Information)*

[RFC2974] M. Handley, C. Perkins, E. Whelan, "Protocole d'annonce de session (SAP)", octobre 2000. *(Expérimentale)*

[RFC3048] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd et M. Luby, "Blocs de construction de transport fiable en diffusion groupée pour transfert de données en vrac en point à multipoint", janvier 2001. *(Info.)*

[RFC3269] R. Kermode et L. Vicisano, "Lignes directrices pour les auteurs de documents de mise en œuvre de protocole et de blocs de construction de transport fiable en diffusion groupée (RMT)", avril 2002.

[RFC3451] M. Luby et autres, "Bloc de construction du transport de codage en couches (LCT)", décembre 2002. *(Remplacée par* RFC5651)

[RFC3453] M. Luby et autres, "Utilisation de la correction d'erreur directe (FEC) en diffusion groupée fiable", décembre 2002. *(Info.)*

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications en temps réel", STD 64, DOI 10.17487/RFC3550, juillet 2003. *(MàJ par* RFC7164, RFC7160, RFC8083, RFC8108, RFC8860)

[RFC3738] M. Luby, V. Goyal, "Bloc de construction de contrôle de débit fondé sur l'onde et l'équation (WEBRC)", avril 2004. *(Exp.)*

[RFC4566] M. Handley, V. Jacobson et C. Perkins, "SDP : Protocole de description de session", juillet 2006, DOI 10.17487/RFC4566. *(P.S. ; remplacée par RFC8866)*

[RFC4607] H. Holbrook, B. Cain, "Diffusion groupée spécifique de source pour IP", août 2006. *(P.S.)*

[RFC5775] M. Luby, M. Watson, L. Vicisano, "Instance de protocole de codage en couches asynchrone (ALC)", avril 2010. (*Remplace* RFC3450). *(P. S.)*

[RIZ1997a] Rizzo, L., "Effective Erasure Codes for Reliable Computer Communication Protocols", ACM SIGCOMM Computer Communication Review, Vol.27, No.2, pp.24-36, avril 1997.

[RIZ1997b] Rizzo, L. and L. Vicisano, "Reliable Multicast Data Distribution protocol based on software FEC techniques", Proceedings of the Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems, HPCS'97, Chalkidiki Greece, juin 1997.

[RIZ2000] Rizzo, L., "PGMCC: A TCP-friendly single-rate multicast congestion control scheme", Proceedings of SIGCOMM 2000, Stockholm Sweden, août 2000.

[VIC1998] Vicisano, L., Rizzo, L., and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer", IEEE Infocom'98, San Francisco, CA, mars 1998.

Adresse des auteurs

Michael Luby
Qualcomm, Inc.
3165 Kifer Rd.
Santa Clara, CA 95051
US
mél : luby@qualcomm.com

Mark Watson
Qualcomm, Inc.
3165 Kifer Rd.
Santa Clara, CA 95051
US
mél : watson@qualcomm.com

Lorenzo Vicisano
Qualcomm, Inc.
3165 Kifer Rd.
Santa Clara, CA 95051
US
mél : vicisano@qualcomm.com