

Groupe de travail Réseau
Request for Comments : 2687
Catégorie : En cours de normalisation

C. Bormann, Universitaet Bremen TZI
septembre 1999
Traduction Claude Brière de L'Isle

PPP dans une trame de style HDLC orientée temps réel

Statut du présent 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 (1999). Tous droits réservés.

Résumé

Un document voisin décrit une architecture de fourniture de services intégrés sur des liaisons à faible débit binaire, comme les lignes de modem, les canaux B du RNIS, et les liaisons sub-T1 de la [RFC2689]. Les principaux composants de l'architecture sont un format d'encapsulation en temps réel pour les liaisons asynchrones et synchrones à faible débit binaire, une architecture de compression d'en-tête optimisée pour les flux en temps réel, les éléments des protocoles de négociation utilisés entre les routeurs (ou entre les hôtes et les routeurs) et des protocoles d'annonces utilisés par les applications pour permettre à cette négociation d'avoir lieu.

Le présent document propose la solution orientée suspension/reprise pour la partie format d'encapsulation en temps réel de l'architecture. L'approche générale est de partir du protocole de fragmentation multi liaison de PPP [RFC1990] et son extension multi-classes [RFC2686] et d'ajouter la suspension/reprise d'une façon qui soit aussi compatible que possible avec les matériels et logiciels existants.

1. Introduction

Comme extension aux services "au mieux" qui sont bien connus dans l'Internet, des types supplémentaires de services ("services intégrés") qui prennent en charge le transport d'informations multimédia en temps réel sont en train d'être développés pour, et déployés dans, l'Internet.

Le présent document définit la solution orientée suspension/reprise pour la partie format d'encapsulation en temps réel de l'architecture. Comme décrit plus en détails dans le document d'architecture, un format d'encapsulation en temps réel est exigé, car, par exemple, un paquet de 1500 octets sur une liaison modem à 28,8 kbit/s rend cette liaison indisponible pour la transmission d'informations en temps réel pendant environ 400 ms. Cela ajoute un pire cas de délai qui est cause que les applications en temps réel fonctionnent avec des délais d'aller-retour de l'ordre d'au moins une seconde – ce qui est inacceptable pour une conversation en temps réel.

Une approche vraiment orientée suspension/reprise peut seulement être mise en œuvre sur un expéditeur de type-1 [RFC2689], mais fournit les meilleures performances de délai possibles pour ce type d'expéditeurs. Le format défini dans le présent document peut aussi intéresser certains expéditeurs de type-2 qui veulent exploiter la meilleure efficacité binaire de ce format comparée à celle de la [RFC2686]. Le format a été conçu de façon qu'il puisse être mis en œuvre par les récepteurs aussi bien de type-1 que de type-2.

1.1 Langage de spécification

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

2. Exigences

Les exigences pour le présent document sont similaires de celles énumérées dans la [RFC2686].

Une solution orientée suspension/reprise peut fournir un meilleur pire cas de latence que la solution orientée pré fragmentation définie dans la [RFC2686]. Aussi, comme cette solution exige un nouveau schéma d'encapsulation, il y a une opportunité de fournir un format légèrement plus efficace.

Prévisibilité, robustesse, et coopération avec PPP et les installations existantes de matériels et logiciels sont aussi importantes avec la suspension/reprise qu'avec la pré fragmentation. Une bonne solution de suspension/reprise réalise de bonnes performances même avec les receveurs de type-2 [RFC2689] et est capable de fonctionner avec des matériels PPP tels que les convertisseurs d'asynchrone en synchrone.

Finalement, une non exigence partielle : alors que le format défini dans ce projet se fonde sur le protocole PPP multi-liaison ([RFC1990], aussi abrégé en MP) le fonctionnement sur plusieurs liaisons n'est dans de nombreux cas pas exigé.

3. Approche générale

Comme dans la [RFC2686], l'approche générale est de commencer à partir d'une multi-liaison PPP et d'ajouter plusieurs classes pour obtenir plusieurs niveaux de suspension. Cependant, à la différence de la [RFC2686], des changements plus significatifs sont requis pour être capable de suspendre la transmission d'un paquet à tout moment et d'injecter un paquet de priorité supérieure.

L'applicabilité de l'en-tête multi-liaison pour les mises en œuvre de type suspension/reprise est limitée, car le bit "fin" est dans l'en-tête multi-liaison, qui est le mauvais endroit pour le fonctionnement de suspension/reprise. Pour rendre possible la suspension d'un gros paquet, il doit être envoyé avec le bit "fin" à zéro, et (sauf si le paquet a été suspendu un petit nombre d'octets avant sa fin) un fragment vide va devoir être envoyé ensuite pour "clore" le paquet. La redondance minimale pour l'envoi d'un paquet suspendible est donc de deux fois la taille de l'en-tête multi-liaison (six octets, incluant un champ de protocole multi-liaison compressé) plus un tramage PPP (trois octets). Chaque suspension coûte six autres octets (non comptée la redondance pour le tramage du paquet intercalé).

Aussi, l'en-tête multi-liaison existant est relativement grand, car lorsque la fréquence des petits paquets de forte priorité augmente, la redondance devient significative.

L'approche générale du présent document est de commencer à partir d'une multi-liaison PPP avec des classes et de fournir un certain nombre d'extensions pour ajouter des fonctionnalités et réduire la redondance de l'utilisation de la multi-liaison PPP pour la transmission en temps réel.

Le présent document introduit deux nouvelles caractéristiques :

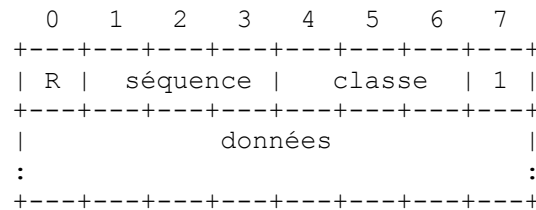
- 1) un format et en-tête de fragment compact,
- 2) un format de trame de temps réel.

4. Format de fragment compact

La présente section décrit un format facultatif de fragment multi-liaison qui est plus optimisé à l'égard du fonctionnement sur une seule liaison et une fréquente suspension (envoyeurs de type 1)/une petite taille de fragment (envoyeurs de type 2) avec prise en charge facultative de liaisons multiples.

En fonctionnement sur une seule liaison, le numéro de séquence multi-liaison n'est utilisé que pour la détection de pertes. Même un numéro de séquence de 12 bits est clairement plus grand que ce qui est requis pour cette application sur la plupart des types de liaisons. On définit donc l'option de format d'en-tête multi-liaison compact suivante avec un numéro de séquence de trois bits.

Comme, avec un en-tête compact, il y a peu de besoins d'envoi de paquets en dehors de la multi-liaison, on peut fournir un mécanisme de compression supplémentaire pour ce format : l'identifiant de protocole MP n'est pas envoyé avec l'en-tête de fragment compact. Cela exigé évidemment une négociation préalable (similaire à la façon dont est négociée la compression des champs d'adresse et de contrôle) ainsi qu'une méthode pour éviter la combinaison binaire 0xFF (le premier octet dans une trame LCP avant que toute option LCP ait été négociée) car autrement, le début d'une nouvelle négociation LCP ne pourrait pas être détecté de façon fiable.

Figure 1 : Format de fragment compact

Avoir le bit de moindre poids toujours à 1 aide avec les jetons HDLC qui opèrent spécialement sur les bits de moindre poids dans les adresses HDLC. (Les octets initiaux avec le bit de moindre poids réglé à zéro sont utilisés pour le format de fragment compact étendu ; voir à la section suivante.)

Le bit R est l'équivalent inversé du bit B dans les autres formats de fragment multi-liaison, c'est-à-dire, R = 1 signifie que ce fragment reprend les précédents fragments d'un paquet qui ont déjà été envoyés.

Le truc suivant évite le cas où un octet d'en-tête de 0xFF (qui devrait signifier R = 1, séquence = 7, et classe = 7) : si le champ de classe est réglé à 7, le bit R NE DOIT JAMAIS être réglé à un. C'est-à-dire que les trames de classe 7 ne peuvent, par conception, être suspendues/reprises. (C'est aussi la raison pour laquelle le sens du bit B est inversé en un bit R dans le format de fragment compact -- classe 7 serait inutile autrement, car un nouveau paquet ne pourrait jamais être commencé.)

Comme le numéro de séquence n'est pas particulièrement utile avec le champ de classe réglé à 7, il est utilisé pour distinguer huit classes de plus -- pour une complexité supplémentaire mineure, l'applicabilité de l'émission de préfixe est significativement augmentée en fournissant plus de classes avec des préfixes élidés éventuellement différents.

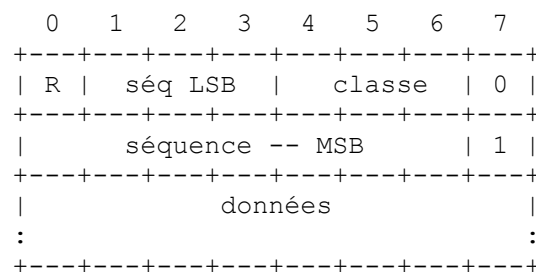
Pour les besoins de l'émission de préfixe, le numéro de classe réel d'un fragment est calculé comme suit :

- si le champ de classe est de 0 à 6, le numéro de classe est de 0 à 6,
- si le champ de classe est 7 et si le champ de séquence est de 0 à 7, le numéro de classe est de 7 à 14.

Il résulte de ce schéma que les classes 0 à 6 peuvent être utilisées pour des paquets suspendibles, et les classes 7 à 14 (où le champ de classe est 7 et où le bit R doit toujours être à zéro) peuvent être utilisée pour des classes à forte priorité non suspendibles, par exemple, huit flux vocaux à haute compression.

5. Format de fragment compact étendu

Pour le fonctionnement sur des liaisons multiples, un numéro de séquence de trois bits va rarement être suffisant. Donc, on définit un format de fragment compact étendu facultatif. L'option, lorsque elle est négociée, permet d'utiliser aussi bien le format de fragment compact de base que le format de fragment compact étendu ; chaque fragment indique dans quel format il est.

Figure 1: Format de fragment compact étendu

Dans le format de fragment compact étendu, le numéro de séquence se compose des trois bits de moindre poids du premier octet de l'en-tête de fragment et des sept bits de poids fort du second octet. (Là encore, le bit de moindre poids du second octet est toujours réglé à un pour la compatibilité avec certains jetons HDLC.)

Pour les besoins de l'émission de préfixe, les fragments avec un champ de classe de 7 peuvent utiliser le format de base pour indiquer les classes 7 à 14 et le format étendu pour indiquer les classes 7 à 1030. Différentes classes peuvent utiliser concurremment différents formats sans problème. (Cela permet que certaines classes soient étalées sur une multi-liaison et

que d'autres classes soient confinées à une seule liaison avec une plus grande efficacité.) Pour les champs de classes 0 à 6, c'est-à-dire les classes suspensibles, un des deux formats de fragment compact DEVRAIT être utilisé de façon cohérente au sein de chaque classe.

Si l'utilisation du format de fragment compact étendu a été négociée, les receveurs PEUVENT garder des numéros de séquence à 10 bits pour toutes les classes pour faciliter aux envoyeurs le changement de formats dans une classe. Lorsque un envoyeur commence à envoyer des fragments au format de basse dans une classe qui utilisait des fragments de format étendu, les trois bits du numéro de séquence peuvent être pris comme une version modulo 8 du numéro de séquence à 10 bits, et aucune discontinuité ne devrait en résulter. Dans le cas inverse, si un numéro de séquence à 10 bits a été conservé tout au long par le receveur (et si aucun glissement majeur du numéro de séquence ne s'est produit) aucune discontinuité ne va en résulter, bien que ceci ne puisse être garanti en présence d'erreurs. (Une discontinuité, dans ce contexte, signifie qu'un receveur doit resynchroniser les numéros de séquence en éliminant les fragments jusqu'à voir un fragment avec R = 0.)

6. Format de trame de temps réel

La présente section définit comment les fragments avec des en-têtes de fragment compact sont transposés en trames en temps réel. Ce format a été conçu pour conserver le format global de trame fondé sur HDLC, afin que les jetons HDLC synchrones existants et les convertisseurs d'asynchrone en synchrone puissent être utilisés sur la liaison. Noter que si la conception pouvait être optimisée pour le fonctionnement seulement en asynchrone, plus de solutions de remplacement seraient disponibles [QOSPPP] ; avec l'arrivée de modems de style V.80, les communications asynchrones vont cependant vraisemblablement diminuer d'importance.

Le format de fragment compact fournit un rendu compact de l'en-tête multi-liaison PPP avec des classes et un espace de numéro de séquence réduit. Cependant, il ne code pas le bit E de l'en-tête multi-liaison PPP, qui indique si le fragment actuel est le dernier fragment d'un paquet.

Pour une solution où les paquets puissent être suspendus à tout instant, le bit E doit être codé près de la fin de chaque fragment. Le format de trame en temps réel, pour assurer la compatibilité maximale avec les receveurs de type 2, code le bit E de la façon suivante : toute terminaison de trame normale termine aussi le fragment actuel avec E implicitement réglé à un. Cela assure que ces paquets qui sont prêts à la livraison à la couche supérieure déclenchent immédiatement une interruption de réception même chez les receveurs de type-2.

Les fragments de paquets qui sont à suspendre se terminent au sein de la trame HDLC par un octet spécial "échappement de fragment suspendu" (FSE, *fragment suspend escape*). La structure globale de la trame HDLC ne change pas ; la détection et le traitement des octets FSE est fait à une couche supérieure au tramage HDLC.

Le format suspension/reprise avec détection de FSE est une solution de remplacement à la compression du champ adresse/contrôle (ACFC, LCP option 8). Il ne s'applique pas aux trames qui commencent par 0xFF, le champ d'adresse standard de PPP dans HDLC ; ces trames sont traitées comme défini dans la [RFC1661] et la [RFC1662]. (Cette disposition assure que les tentatives de renégocier LCP ne causent pas d'ambiguïtés.)

Pour les trames qui ne commencent pas par 0xFF, le traitement de suspension/reprise effectue un examen de chaque trame HDLC reçue. Le FCS de la trame HDLC est vérifié et supprimé. Les en-têtes de format de fragment compact (de base et étendu) sont traités sans autre traitement du FSE. (Noter que, comme l'octet FSE a été choisi de telle façon qu'il ne survienne jamais dans les en-têtes de format de fragment compact, cela n'exige aucun code spécifique.)

Au sein des octets restants de la trame HDLC ("partie données") un octet FSE est utilisé pour indiquer la fin du fragment en cours, avec un bit E implicitement à zéro. Tous les fragments jusqu'au dernier FSE sont considérés suspendus (E = 0) ; le fragment final est terminé (E = 1) ou, si il est vide, ignoré (c'est-à-dire, la partie données d'une trame HDLC peut se terminer dans un FSE pour indiquer que le dernier fragment a E = 0).

Chaque fragment commence par un en-tête normal, de sorte que la structure d'une trame pourrait être :

Figure 2 : Exemple de trame avec délimiteur FSE

```

  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| R | séquence | classe | 1 |
+---+---+---+---+---+---+---+---+
|                               |
:                               :
+---+---+---+---+---+---+---+---+
+                               + précédent fragment implicitement E = 0
+---+---+---+---+---+---+---+---+
| R | séquence | classe | 1 |
+---+---+---+---+---+---+---+---+
|                               |
:                               :
+---+---+---+---+---+---+---+---+
|                               | précédent fragment implicitement E = 1
|                               |
+---+---+---+---+---+---+---+---+

```

La valeur choisie pour FSE est 0xDE (c'est un octet relativement improbable dans les flux de données d'aujourd'hui, il ne déclenche pas de bourrage d'octet et ne déclenche le bourrage de bit que pour 1/8 des octets précédents possibles).

Le problème restant est celui de la transparence des données. Dans le schéma décrit jusqu'à présent, un FSE est toujours suivi par un en-tête de fragment compact. Dans ces en-têtes, la combinaison d'un champ de classe réglé à 7 avec R = 1 est réservée. La transparence des données est réalisée en faisant un cas particulier de l'occurrence d'un octet FSE suivi par un des caractères de 0x8F, 0x9F, ... à 0xFF.

Figure 3 : Transparence des données avec octets FSE présents

```

  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| R | séquence | classe | 1 |
+---+---+---+---+---+---+---+---+
|                               |
:                               :
+---+---+---+---+---+---+---+---+
+                               + fragment NON terminé
+---+---+---+---+---+---+---+---+
| R | S | T | U | 1 | 1 | 1 | 1 | R est toujours 1
+---+---+---+---+---+---+---+---+
|                               | fragment continuant
:                               :

```

Dans une combinaison de FSE/0xnF (où n est le premier champ de quatre bits dans le second octet, RSTU à la Figure 3) le champ n donne une séquence de quatre bits qui indique où dans le flux de données reçu, les octets FSE, qui ne peuvent pas simplement être transmis dans le flux de données, sont à ajouter par le receveur :

0x8F : insérer un FSE, retour aux données

0x9F : insérer un FSE, copier deux octets de données, insérer un FSE, retour aux données

0xAF : insérer un FSE, copier un octet de données, insérer un FSE, retour aux données

0xBF : insérer un FSE, copier un octet de données, insérer deux octets de FSE, retour aux données

0xCF : insérer deux octets de FSE, retour aux données

0xDF : insérer deux octets de FSE, copier un octet de données, insérer un FSE, retour aux données

0xEF : insérer trois octets de FSE, retour aux données

0xFF : insérer quatre octets de FSE, retour aux données

Les octets de données qui suivent la combinaison FSE/0xnF et correspondent aux bits zéro dans le champ N ne peuvent pas être des octets de FSE.

Ce schéma limite le pire cas de facteur d'expansion par le traitement de FSE d'environ 25 %. Il est aussi conçu de façon à ce qu'un seul flux de données puisse déclencher le pire cas d'expansion par bourrage d'octet (ou par bourrage de bit) ou le pire cas de traitement de FSE, mais jamais les deux. La Figure 4 illustre le schéma par quelques exemples ; les paires FSE/0xnF sont écrites en minuscules.

Figure 4 : Exemple de transparence des données

Flux de données	Flux bourré FSE
DD DE DF E0	DD de 8f DF E0
01 DE 02 DE 03	01 de af 02 03
DE DA DE DE DB	de bf DA DB
DE DE DE DE DE DA	de ff de 8f DA

En résumé, le format de trame en temps réel est une trame de style HDLC délimitée par des fanions et contenant un FCS final comme défini dans la [RFC1662], mais sans les champs d'adresse et de contrôle, contenant comme données une séquence de fragments de FSE bourrés en format de fragment compact, délimités par des octets de FSE. Comme cas particulier, le FSE final peut survenir comme le dernier octet du contenu de données (c'est-à-dire, immédiatement avant les octets FCS) de la trame de style HDLC, pour indiquer que le dernier fragment de la trame est suspendu et qu'aucun fragment final n'est dans la trame (par exemple, parce que la taille maximum désirable de la trame a été atteinte).

7. Notes de mise en œuvre

7.1 Questions de MRU

Le paramètre LCP MRU définit la taille maximum des paquets envoyés sur la liaison. Les convertisseurs d'asynchrone en synchrone qui surveillent les négociations LCP sur la liaison peuvent interpréter la valeur de la MRU comme la taille maximum de trame HDLC à attendre.

Les mises en œuvre de la présente spécification devraient de préférence négocier une MRU suffisamment grande pour couvrir le pire cas de 25 % d'augmentation de la taille de trame plus l'augmentation causée par les fragments suspendus. Si cela n'est pas possible, la taille de trame HDLC devrait être limitée par la surveillance des tailles de trames HDLC et éventuellement la suspension du fragment en cours par l'envoi d'un FSE avec un fragment final vide (FSE immédiatement suivi par la fin du champ d'information, c'est-à-dire, par les octets de CRC et un fanion) pour être capable de continuer dans une nouvelle trame HDLC. Cette stratégie aide aussi à minimiser l'impact de l'allongement de la trame HDLC sur la sécurité des 16 bits de FCS à la fin de la trame HDLC.

7.2 Mise en œuvre du bourrage d'octet et du traitement FSE dans un automate

La façon la plus simple pour ajouter un tramage en temps réel à une mise en œuvre peut être d'effectuer le traitement HDLC comme d'habitude, puis avec le résultat, d'effectuer le traitement FSE. Une mise en œuvre plus évoluée peut vouloir combiner les deux niveaux de traitement de caractère d'échappement. Noter cependant que le traitement de FSE doit attendre que deux octets de la trame HDLC soient disponibles et soient suivis par un troisième pour assurer que les octets ne sont pas les octets finaux de FCS de l'HDLC, qui ne sont pas sujets au traitement FSE. C'est-à-dire qu'à réception d'un octet de données normal, on cherche un FSE dans le second octet précédent et à réception d'une fin de trame, on cherche un FSE comme dernier octet de données.

8. Options négociables

Les options suivantes sont déjà définies par MP [RFC1990] :

- o Unité maximum reconstruite reçue en multi-liaison
- o Format d'en-tête de numéro de séquence court de multi-liaison
- o Discriminateur de point d'extrémité

Les options suivantes sont déjà définies par MCML [RFC2686]:

- o Format d'en-tête multi-liaison
- o Élision de préfixe

Le présent document définit deux nouveaux codets pour l'option Format d'en-tête multi-liaison.

8.1 Option de format d'en-tête multi-liaison

L'option Format d'en-tête multi-liaison est définie dans la [RFC2686]. Un résumé du format de l'option Format d'en-tête

multi-liaison est donné ci-dessous. Les champs sont transmis de gauche à droite.

Figure 5 : Option Format d'en-tête multi-liaison

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Type = 27   | Longueur = 4   |      Code      |Nbr classe susp|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Comme défini dans la [RFC2686], cette option LCP indique à l'homologue que la mise en œuvre souhaite recevoir des fragments dans un format donné par le numéro de code, avec le nombre maximum de classes de suspension (voir ci-dessous) indiqué. La présente spécification définit deux valeurs supplémentaires de code, en plus de celles définies dans la [RFC2686] :

- Code = 11 : format de base et étendu de fragment compact en temps réel avec des classes, en trame HDLC codée en FSE
- Code = 15 : format de base de fragment compact en temps réel avec classes, en trame HDLC codée en FSE

Une mise en œuvre NE DOIT PAS demander une combinaison des deux LCP Compression de champ d'adresse et de contrôle (ACFC, *Address-and-Control-Field-Compression*) et des valeurs de code 11 ou 15 pour cette option.

Le nombre de classes suspensibles négociées pour le format compact de fragment en temps réel limite seulement l'utilisation des numéros de classe qui permettent la suspension. Comme les numéros de classe de 7 et au dessus n'exigent pas d'espace supplémentaire de réassemblage, ils ne sont pas soumis à la limitation de numéros de classe négociée.

9. Considérations pour la sécurité

Le fonctionnement de ce protocole est estimé n'être ni plus ni moins sûr que le fonctionnement du protocole PPP multi-liaison [RFC1990]. Le fonctionnement avec une petite gamme de numéros de séquence augmente la probabilité que des fragments provenant de paquets différents puissent être incorrectement réassemblés dans un seul paquet. Bien que la plupart de tels paquets soient éliminés par le receveur à cause d'échecs de somme de contrôle de couche supérieure ou autres incohérences, il y a une probabilité accrue que le contenu des paquets destinés à un hôte puisse être livré à un autre hôte. Les liaisons qui portent des paquets où cela soulève des problèmes de sécurité DEVRAIENT utiliser la gamme étendue de numéros de séquence pour les paquets multi fragments.

10. Références

- [RFC1661] W. Simpson, éditeur, "[Protocole point à point](#) (PPP)", STD 51, juillet 1994. (*MàJ par la RFC2153*)
- [RFC1662] W. Simpson, éditeur, "[PPP en trames de style HDLC](#)", STD 51, juillet 1994. (*Remplace la RFC1549*)
- [RFC1973] W. Simpson, "[PPP en relais de trame](#)", juin 1996. (*P.S.*)
- [RFC1990] K. Sklower et autres, "Protocole [multiliaison en PPP](#) (MP)", août 1996. (*Remplace RFC1717*) (*D.S.*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2686] C. Bormann, "[Extension Multi-classe à PPP](#) multi-liaison", septembre 1999. (*P.S.*)
- [RFC2689] C. Bormann, "Fourniture de services intégrés sur liaisons à faible débit", septembre 1999, (*Information*)
- [QOSPPP] Andrades, R. and F. Burg, "QOSPPP Framing Extensions to PPP", (*Non publiée comme RFC*)

11. Adresse de l'auteur

Carsten Bormann
Universitaet Bremen FB3 TZI
Postfach 330440
D-28334 Bremen, GERMANY

téléphone : +49.421.218-7024
fax : +49.421.218-7000
mél : cabo@tzi.org

Remerciements

Les participants à un BOF déjeuner à la réunion de Montréal de l'IETF de 1996 ont fait des apports utiles sur les compromis de conception dans divers environnements. Richard Andrades, Fred Burg, et Murali Aravamudan ont insisté pour qu'il y ait une solution suspension/reprise en plus de celle de pré-fragmentation définie dans la [RFC2686]. Les membres du sous-groupe ISSLL sur les liaisons à faible débit binaire (ISSLOW) ont aidé à réaliser un ensemble d'exigences qui ont donné forme à cette solution.

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1999). Tous droits réservés.

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

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

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

Remerciement

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