

Groupe de travail Réseau
Request for Comments : 2728
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

R. Panabaker, Microsoft
S. Wegerif, Philips Semiconductors
D. Zigmund, WebTV Networks
novembre 1999

Transmission de IP sur l'intervalle de suppression vertical d'un signal de télévision

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.

1. Résumé

Le présent document décrit une méthode pour diffuser des données IP de façon unidirectionnelle en utilisant l'intervalle de suppression vertical des signaux de télévision. Il inclut une description de la compression des en-têtes IP sur des réseaux unidirectionnels, un protocole de tramage identique à SLIP, un schéma de correction d'erreur directe, et les structures d'octet NABTS.

2. Introduction

La présente RFC propose plusieurs protocoles à utiliser pour la transmission de datagrammes IP utilisant l'intervalle de suppression vertical (VBI, *Vertical Blanking Interval*) d'un signal de télévision. Le VBI est une portion non visible du signal de télévision qui peut être utilisée pour fournir des services de données IP en point à multipoint (en diffusion) qui vont soulager l'encombrement et le trafic dans les réseaux d'accès traditionnels de l'Internet. Chaque fois que possible, ces protocoles utilisent les RFC standard et non standard existantes.

Traditionnellement, les connexions point à point (TCP/IP) ont été utilisées même pour la transmission de données de type diffusion. La distribution du même contenu -- nouvelles, cotations boursières, groupes de nouvelles, rapports météorologiques, et ainsi de suite -- est normalement envoyée de façon répétée à des clients individuels plutôt que diffusée aux usagers qui en grand nombre veulent recevoir de telles données.

Aujourd'hui, IP devient rapidement la méthode préférée de distribution de données d'un point à de nombreux autres sur les intranets et l'Internet. La nouvelle disponibilité de matériels PC à bas prix pour recevoir des signaux de télévision accompagnés par des flux de données diffusées rend impérative la définition d'une norme pour la transmission de données sur les réseaux de diffusion traditionnels. L'absence d'une norme dans ce domaine ainsi que le coût des matériels a empêché les réseaux de diffusion traditionnels de devenir le moyen de livraison effectif des données au domicile et au bureau.

Le présent document décrit la transmission de IP en utilisant la norme nord américaine de télétexte de base (NABTS, *North American Basic Teletext Standard*) méthode reconnue et soutenue par l'industrie du transport de données sur le VBI. NABTS est traditionnellement utilisé sur les systèmes de télévision à 525 lignes comme NTSC. Une autre structure d'octets, WST, est traditionnellement utilisée sur les systèmes à 625 lignes comme le PAL et le SECAM. Ces généralisations souffrent des exceptions, et chaque pays devrait être considéré individuellement. Ces normes de système de télévision existantes permettent aux communautés de la télévision et de l'Internet de fournir des services de diffusion de données très bon marché. Un ensemble de protocoles existants seront mis en couche par dessus la FEC spécifique de NABTS incluant un protocole de tramage de flux de série similaire à SLIP (RFC1055 [15]) et une technique de compression pour les en-têtes UDP/IP unidirectionnels.

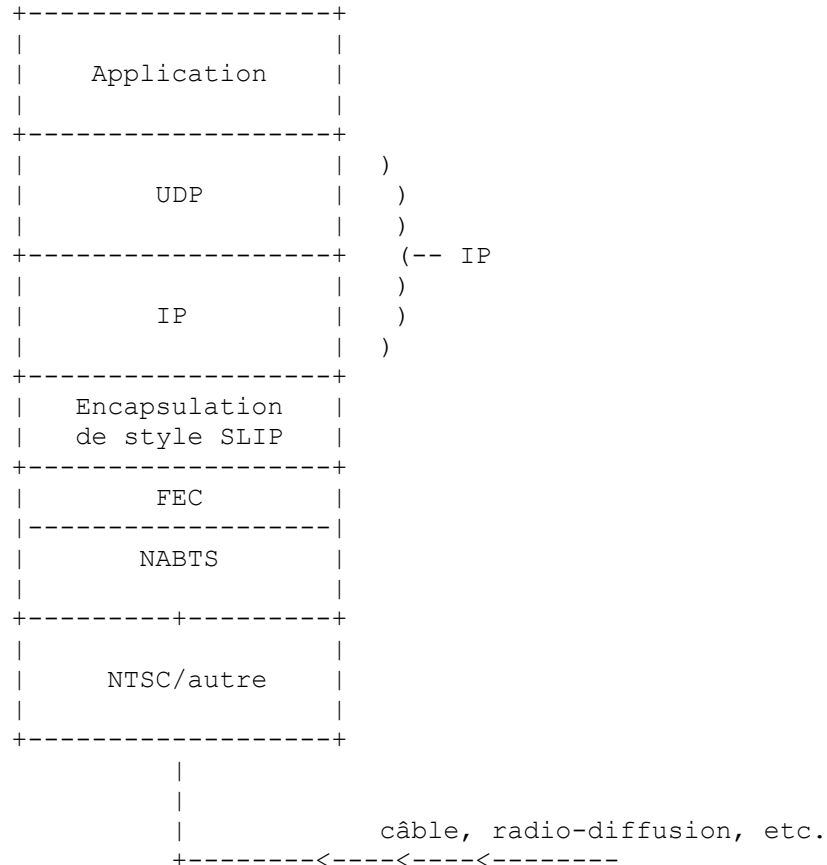
Les protocoles décrits dans le présent document sont destinés à la livraison unidirectionnelle de datagrammes IP en utilisant le VBI. C'est-à-dire qu'aucun canal de retour n'est décrit, ni possible, dans le VBI.

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 RFC 2119.

3. Proposition de pile de protocole

La pile de protocoles suivante montre les couches utilisées dans la transmission des données VBI. Chaque couche n'a aucune connaissance des données qu'elle encapsule, et est donc une notion abstraite pour les autres couches. À la couche liaison, le protocole NABTS définit le schéma de modulation utilisé pour transporter les données sur le VBI. À la couche réseau, IP traite le mouvement des données vers le client approprié. Dans la couche transport, UDP détermine le flux de données vers les processus et applications appropriés.



Ces protocoles peuvent être décrits dans un modèle de composants de bas en haut en utilisant l'exemple de NABTS porté sur la modulation NTSC comme suit :

Signal vidéo ---> NABTS ---> FEC ---> flux de données en série ---> protocole de tramage ---> en-têtes UDP/IP compressés --> données d'application

Ce diagramme peut être lu comme suit : les signaux de télévision ont des paquets NABTS, qui contiennent un protocole de correction d'erreur directe (FEC) modulés sur eux. Les données contenues dans cette suite ordonnée de paquets forment un flux de données sériel sur lequel un protocole de tramage indique la situation des paquets IP, avec les en-têtes compressés qui contiennent les données d'application.

La structure de ces composants et protocoles est décrite dans les paragraphes qui suivent.

3.1 VBI

Les caractéristiques et la définition du VBI dépend du système de télévision utilisé, qu'il soit le NTSC, le PAL, le SECAM ou un autre. Pour plus d'information sur les normes de télévision dans le monde, voir la référence [12].

3.1.1 Systèmes à 525 lignes

Une trame de télévision de 525 lignes se compose de deux champs de 262,5 lignes d'examen horizontales chacun. Les 21 premières lignes de chaque champ ne font pas partie de l'image visible et sont collectivement appelées l'intervalle de suppression vertical (VBI, *Vertical Blanking Interval*).

De ces 21 lignes, les 9 premières sont utilisées lors du repositionnement de la raie cathodique au sommet de l'écran, mais les lignes restantes sont disponibles pour le transport de données.

Il y a 12 lignes de VBI possibles qui sont diffusées 60 fois par seconde (chaque champ 30 fois par seconde). Dans certains pays, la ligne 21 est réservée pour le transport de données de sous-titrage (réf. [11]). Dans ce cas, il y a 11 lignes de VBI possibles, dont certaines ou toutes pourraient être utilisées pour le transport IP. On notera que certaines de ces lignes sont parfois utilisées pour des services existants, propriétaires, de données et d'essais. La livraison IP devient donc un service de données de plus qui utilise un sous ensemble de ces lignes ou toutes.

3.1.2 Systèmes à 625 lignes

Une trame de télévision à 625 lignes se compose de deux champs de 312,5 lignes d'examen horizontales chacun. Les premières lignes de chaque champ sont utilisées lors du repositionnement de la raie cathodique au sommet de l'écran. Les lignes disponibles pour l'insertion de données sont les lignes 6 à 22 dans le premier champ et 319 à 335 dans le second.

Il y a donc 17 lignes de VBI possibles qui sont diffusées 50 fois par seconde (chaque champ 25 fois par seconde) dont certaines, ou toutes, pourraient être utilisées pour le transport IP. On notera que certaines de ces lignes sont parfois utilisées pour des services existants, propriétaires, de données et d'essais. IP devient donc un service de données de plus qui utilise un sous ensemble de ces lignes, ou toutes.

3.2 NABTS

La norme nord américaine de télétexte de base est définie dans le document EIA-561 de l'Association des industries électroniques [2], et dans la Recommandation BT.653-2, "système C", réf. [13] de l'UIT-R. Elle fournit une méthode acceptée par l'industrie pour moduler des données sur le VBI, usuellement d'un signal NTSC. La présente section décrit le format du paquet NABTS comme il est utilisé pour le transport de IP.

On notera que seul un sous ensemble de la norme NABTS est utilisé, comme il est de pratique courante dans les mises en œuvre de NABTS. D'autres informations concernant la norme NABTS et sa mise en œuvre se trouvent dans EIA-516.

Le paquet NABTS est une structure de 36 octets codée sur une ligne d'examen horizontale d'un signal de télévision qui a la structure suivante :

```

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
+-----+-----+-----+-----+-----+-----+-----+-----+
| Synchronisation d'horloge | Sync. d'octet | Adresse paquet |
+-----+-----+-----+-----+-----+-----+-----+
| Adresse paquet (suite)   | Indice contin. | FanionsStuctPqt |
+-----+-----+-----+-----+-----+-----+-----+
|                               Données d'utilisateur (26 octets)                               |
:                                                                           :
:                                                                           :
|                               +-----+-----+-----+-----+                               |
|                               |                               |                               |
|                               |                               |                               |
|                               |                               |                               |
+-----+-----+-----+-----+-----+-----+-----+

```

Les deux octets de synchronisation d'horloge et un octet de synchronisation d'octet sont situés au début de chaque ligne d'examen contenant un paquet NABTS et ils sont utilisés pour synchroniser le taux d'échantillonnage de décodage et l'écoulement des octets.

Les trois octets du champ Adresse de paquet sont codés en Hamming (comme spécifié dans EIA-516) et fournissent 4 bits de données par octet et fournissent donc 4096 adresses de paquet possibles. Ces adresses sont utilisées pour distinguer les différents services générés par la même source. Cela est nécessaire pour que le receveur détermine quels paquets se rattachent à un même service. Les adresses des paquets NABTS distinguent donc les différents services de données, éventuellement insérés à différents points du système de transmission, et sont très probablement totalement décorréliées. La Section 4 de ce document discute en détails des adresses de paquet.

Le champ d'un octet Indice de continuité est codé en Hamming, et il est incrémenté de un pour chaque paquet suivant d'une certaine adresse de paquet. La valeur ou le numéro de l'indice de continuité va de 0 à 15. Il s'incrémente de un chaque fois qu'est transmis un paquet de données. Cela permet au décodeur de déterminer si des paquets ont été perdus durant la transmission.

Le champ Structure de paquet est aussi un octet codé en Hamming, qui contient des informations sur la structure des portions restantes du paquet. Le bit de moindre poids est toujours "0" dans cette mise en œuvre. Le second bit de moindre poids spécifie si le bloc de données est plein -- "0" indique que le bloc de données est plein de données utiles, et "1" indique que certaines des données ou toutes sont du remplissage. Le deux bits de poids fort sont utilisés pour indiquer la longueur du suffixe du bloc de données -- dans cette mise en œuvre, 2 ou 28 octets (10 pour le suffixe de FEC de 2 octets, 11 pour le suffixe de FEC de 28 octets). Ce suffixe est utilisé pour la correction d'erreur directe décrit au paragraphe suivant. Le tableau qui suit montre les valeurs possibles pour le champ Structure de paquet:

Paquet de données, sans remplissage	D0
Paquet de données, avec remplissage	8C
Paquet de FEC	A1

Le champ Bloc de données fait 26 octets, dont de zéro à 26 sont des données utiles (partie d'un paquet IP ou d'une trame SLIP) le reste étant des données de remplissage. Les données sont ordonnées avec le bit de moindre poids en premier. Les données de remplissage sont indiquées par un Ox15 suivi par autant de OxEA qu'il est nécessaire pour remplir le champ Bloc de données. Les blocs de données séquentielles moins les données de remplissage forment un flux sériel asynchrone de données.

Ces paquets NABTS sont modulés en séquence sur le signal de télévision et sur toute combinaison de lignes.

3.3 FEC

À cause de la nature unidirectionnelle du transport de données sur VBI, la correction d'erreur directe (FEC, *Forward Error Correction*) est nécessaire pour assurer l'intégrité des données chez le receveur. Le type de FEC, décrite ici et dans l'appendice au présent document pour NABTS, a été utilisé pendant plusieurs années, et s'est révélé populaire auprès de l'industrie de la diffusion. Elle est capable de corriger des erreurs d'un seul octet et des écrasement sur un seul et sur deux octets dans le bloc et le suffixe de données d'un paquet NABTS. Dans un système qui utilise NABTS, l'algorithme de FEC partage un flux séquentiel de données en "ballots" de 364 octets de données. Les données sont arrangées en 14 paquets de 26 octets chacun. Une fonction est appliquée aux 26 octets de chaque paquet pour déterminer les deux octets de suffixe, qui (additionnés d'un en-tête) complètent le paquet NABTS (8+26+2).

Tous les 14 paquets dans le ballot, deux paquets supplémentaires sont ajoutés, ne contenant que les données de FEC, dont chacun contient 28 octets de données de FEC. Le premier paquet du ballot a une valeur d'Indice de continuité de 0, et les deux paquets de FEC seule à la fin ont des valeurs d'indice de continuité respectivement de 14 et 15. Ces données sont obtenues en écrivant d'abord les paquets dans un tableau de 16 rangées et 28 colonnes.

La même expression que ci-dessus devrait être utilisée dans les colonnes du tableau avec le suffixe maintenant représenté par les octets à la base des colonnes dans les rangées 15 et 16. Avec les en-têtes NABTS sur chacune de ces rangées, on a maintenant un ballot de 16 paquets NABTS prêts pour la modulation séquentielle sur les lignes du signal de télévision.

Chez le receveur, ces formules peuvent être utilisées pour vérifier la validité des données avec une très grande précision. Si on trouve des erreurs sur un seul bit, des erreurs sur un double bit, des erreurs d'un seul octet ou des écrasements d'un seul et d'un double octet dans un rangée ou une colonne (incluant la perte d'un paquet entier) elles peuvent être corrigées en utilisant les algorithmes qui se trouvent dans l'appendice. Le succès de la correction des erreurs va dépendre de la mise en œuvre particulière utilisée chez le receveur.

3.4 Tramage

Un protocole de tramage identique à SLIP est proposé pour encapsuler les paquets décrits dans le paragraphe suivant, faisant donc abstraction des données provenant des couches de protocole inférieures. Ce protocole utilise deux caractères spéciaux : END (0xc0) et ESC (0xdb). Pour envoyer un paquet, l'hôte va envoyer le paquet suivi par le caractère END. Si un octet de données dans le paquet a le même code que le caractère END, une séquence de deux octets de ESC (0xdb) et 0xdc est envoyée à la place. Si un octet de données a le même code que le caractère ESC, une séquence de deux octets de ESC (0xdb) et 0xdd est envoyée à la place. Les mises en œuvre de SLIP sont largement disponibles ; voir les détails dans la RFC1055 [15].

```

+-----+-----+-----+-----+
| paquet |c0| paquet |db|dd| |c0|
+-----+-----+-----+-----+
                        END          ESC          END

```

Le paquet tramé de cette manière devra être formaté conformément à son type de schéma identifié par le champ Schéma, qui doit commencer chaque paquet :

```

+-----+-----+-----+-----+
| Schéma | paquet (formaté selon le schéma) |
| 1 octet | ?? octets (longueur selon le schéma) |
+-----+-----+-----+-----+

```

Dans le cas où le bit de poids fort dans le champ est réglé à "1", la longueur du champ s'étend sur deux octets, permettant 32 768 schémas supplémentaires :

```

+-----+-----+-----+-----+
| Schéma | paquet (formaté selon le schéma) |
| étendu | ?? octets (longueur selon le schéma) |
| 2 octets | |
+-----+-----+-----+-----+

```

Au paragraphe 3.5, on décrit un tel schéma pour l'envoi IP. C'est le seul schéma spécifié dans le présent document. Des schémas supplémentaires peuvent être proposés pour d'autres types de paquets ou d'autres schémas de compression comme décrit à la section 7.

3.4.1 Taille d'unité maximum de transmission

La longueur maximale d'un paquet IP non compressé, ou unité maximum de transmission (MTU, *Maximum Transmission Unit*) est de 1500 octets. Les paquets de plus de 1500 octets DOIVENT être fragmentés avant transmission, et l'interface de client VBI DOIT être capable de recevoir de pleines transmissions de paquets de 1500 octets.

3.5 Schéma de compression d'en-tête IP

Le schéma à un octet définit le format pour le codage du paquet IP lui-même. À cause de la valeur de la bande passante, il peut être souhaitable d'introduire autant d'efficacité que possible dans ce codage. Un des éléments d'une telle efficacité est la compression facultative de l'en-tête UDP/IP en utilisant une méthode qui se rapporte à la compression d'en-tête TCP/IP telle que décrite par Van Jacobson dans la RFC1144 [7]. La compression d'en-tête UDP/IP n'est pas identique à cause des limitations de la transmission unidirectionnelle. Un tel schéma est proposé dans ce document pour la compression de UDP/IP version 4. Il lui est alloué une valeur de 0x00. Tous les schémas futurs d'encapsulation devraient utiliser une valeur unique dans ce champ.

Seul le schéma 0x00 est défini dans ce document ; il doit être accepté par tous les receveurs. Dans le schéma 0x00, le format du paquet IP lui-même prend une parmi deux formes. Les paquets peuvent être envoyés avec des en-têtes pleins, non compressés, comme suit :

```

      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
+-----+-----+-----+-----+
|0|  Groupe | En-tête IP non compressé (20 octets) |
+-----+-----+-----+-----+
|
:                               :
+-----+-----+-----+-----+
| | En-tête UDP non compressé (8 octets) |
+-----+-----+-----+-----+
| | Charge utile (<1472 octets) |
+-----+-----+-----+-----+
|
:                               :
+-----+-----+-----+-----+
CRC |
+-----+-----+-----+-----+

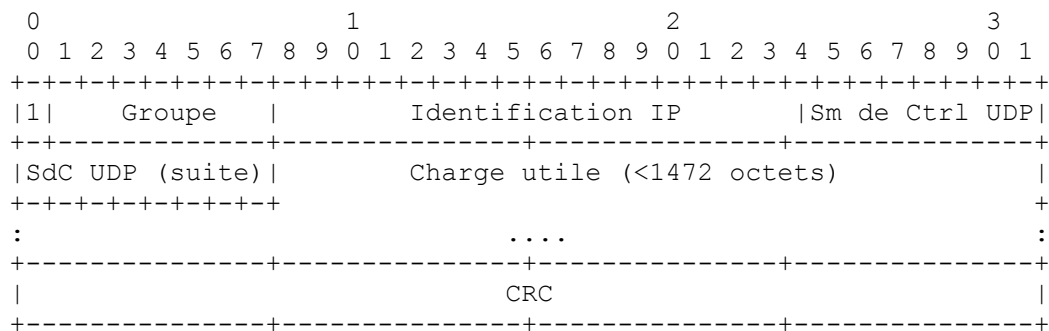
```

Le premier octet du schéma 0x00 est la clé de compression. C'est une valeur d'un octet : le premier bit indique si l'en-tête a été compressé, et les sept bits restants indiquent le groupe de compression auquel il appartient.

Si le bit de poids fort de la clé de compression est réglé à zéro, aucune compression n'est effectuée et l'en-tête complet est envoyé, comme montré ci-dessus. L'interface de client VBI devrait mémoriser le plus récent en-tête non compressé pour une certaine valeur de groupe pour une utilisation future potentielle à la reconstruction des en-têtes compressés suivants. Les paquets avec des bits de groupe identiques sont supposés avoir des en-têtes UDP/IP identiques pour tous les champs UDP et IP, à l'exception des champs "Identification IP" et "Somme de contrôle UDP".

Les valeurs de groupe peuvent être recyclées après 60 secondes de non utilisation par la précédente session UDP/IP (pas de paquet non compressé envoyé) ou en envoyant des paquets avec des en-têtes non compressés pendant la période de 60 secondes suivant le dernier paquet dans la précédente session UDP/IP. De plus, le premier paquet envoyé après 60 secondes de non utilisation, ou une utilisation de seuls paquets à en-tête compressé, doit utiliser un en-tête non compressé. Les interfaces de client VBI ne devraient pas prendre en considération les paquets compressés reçus 60 secondes ou plus après le dernier paquet non compressé utilisant une certaine adresse de groupe. Cela évite les paquets décompressés de façon incorrecte à cause d'une réutilisation de numéro de groupe, et limite les pannes dues à une perte de paquet non compressé jusqu'à 60 secondes.

Si le bit de poids fort de la clé de compression est réglé à un, une version compressée de l'en-tête UDP/IP est envoyée. L'interface de client VBI doit alors combiner l'en-tête compressé avec l'en-tête non compressé mémorisé du même groupe et recréer un paquet complet. Pour les paquets compressés, les seules portions de l'en-tête UDP/IP envoyées sont les champs "Identification IP" et "Somme de contrôle UDP".



Tous les datagrammes qui appartiennent à un paquet IP multi fragments devront être envoyés avec des en-têtes complets, en format d'en-tête non compressé. Donc, seuls les paquets qui n'ont pas été fragmentés peuvent être envoyés avec le bit de poids fort de la clé de compression réglé à "1".

Un CRC de 32 bits a aussi été ajouté à la fin de chaque paquet dans ce schéma pour assurer l'intégrité des données. Il est effectué sur le paquet entier y compris l'octet de schéma, la clé de compression, et les en-têtes compressés ou non compressés. Il utilise le même algorithme que le flux de transport MPEG-2 (ISO/CEI 13818-1). Le polynôme générateur est :

$$1 + D + D2 + D4 + D5 + D7 + D8 + D10 + D11 + D12 + D16 + D22 + D23 + D26 + D32$$

Comme dans la norme ISO/CEI 13818-1, l'état initial de la somme est 0xFFFFFFFF. Ce n'est pas le même algorithme que celui utilisé par Ethernet. Ce CRC donne une vérification finale pour les datagrammes endommagés qui s'étendent sur des ballots de FEC ou qui n'ont pas été bien corrigés par la FEC.

4. Considérations d'adressage

L'adressage des paquets IP devrait suivre les normes existantes du domaine. L'inclusion d'une adresse de source appropriée est nécessaire pour s'assurer que le client receveur peut distinguer les sources et donc les services si plusieurs hôtes partagent un point d'insertion et une adresse de paquet NABTS.

L'adressage du paquet NABTS n'est ni régulé ni normalisé et exige qu'on veuille à s'assurer que des services sans rapport sur le même canal ne sont pas diffusés avec la même adresse de paquet. Cela pourrait arriver du fait de sites d'insertion NABTS multiples éventuels, incluant des productions de spectacles, de la redistribution de réseau, un opérateur régional, et un opérateur local.

Traditionnellement, le marché a reconnu ce problème et a fait des arrangements amiables pour la distribution de ces adresses pour chaque canal.

5. Considérations relatives à l'IANA

L'IANA enregistrera les nouveaux schémas sur la base du "premier arrivé, premier servi" de la [RFC 2434]. Tout le monde peut enregistrer un schéma, pour autant qu'il fournisse un point de contact et une brève description. Le numéro de schéma sera alloué par l'IANA. Les enregistrants sont invités à publier les spécifications complètes des nouveaux schémas (de préférence comme RFC en cours de normalisation) mais ceci n'est pas exigé.

6. Considérations pour la sécurité

Comme avec tout réseau de diffusion, il y a des problèmes de sécurité à cause de l'accessibilité aux données. On suppose que la responsabilité de la sécurité des données incombe aux autres couches de protocole, y compris la suite de protocoles de sécurité IP (IPsec, *IP Security*) les protocoles de sécurité de la couche transport (TLS, *Transport Layer Security*) ainsi que les protocoles de couche application appropriés pour les réseaux exclusivement consacrés à la diffusion.

7. Conclusions

Le présent document donne une méthode pour diffuser des données Internet sur un signal de télévision pour être reçu par les appareils des clients. Avec un modèle de contenu de diffusion approprié, cela va devenir une méthode intéressante pour fournir des services de données aux utilisateurs finaux. En utilisant les normes existantes et une approche de protocoles en couches, le présent document fournit aussi un modèle pour la transmission de données sur les réseaux unidirectionnels et de diffusion.

8. Remerciements

La description de la FEC de l'Appendice A est tirée d'un document préparé par Trevor Dee de Norpak Corporation. Dean Blacketter de WebTV Networks, Inc., a édité le projet final du présent document.

9. Références

- [1] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, RFC2119, mars 1997.
- [2] S. Deering, "Extensions d'hôte pour [diffusion groupée sur IP](#)", STD 5, RFC1112, août 1989. (*Mise à jour par la RFC 2236*)
- [3] EIA-516, "Joint EIA/CVCC Recommended Practice for Teletext: North American Basic Teletext Specification (NABTS)" Washington: Electronic Industries Association, c1988
- [4] Union internationale des télécommunications – secteur des radiocommunications, Recommendation BT.470-5 "Systèmes conventionnels de télévision" (02/98).
- [5] Union internationale des télécommunications – secteur des radiocommunications, Recommendation BT.653-2, "Système C".
- [6] Jack, Keith. "Video Demystified: A Handbook for the Digital Engineer, Second Edition," San Diego: HighText Pub. c1996.
- [7] V. Jacobson, "[Compression des en-têtes TCP/IP](#) pour les liaisons série à faible débit", RFC1144, février 1990.
- [8] Mortimer, Brian. "An Error-correction system for the Teletext Transmission in the Case of Transparent Data" c1989 Department of Mathematics and Statistics, Carleton University, Ottawa Canada

- [9] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, RFC2434, octobre, 1998. (*Rendue obsolète par la RFC5226*)
- [10] Norpak Corporation "TTX71x Programming Reference Manual", c1996, Kanata, Ontario, Canada
- [11] Norpak Corporation, "TES3 EIA-516 NABTS Data Broadcast Encoder Software User's Manual." c1996, Kanata, Ontario, Canada
- [12] Norpak Corporation, "TES3/GES3 Hardware Manual" c1996, Kanata, Ontario, Canada
- [13] Pretzel, Oliver, "Correcting Codes and Finite Fields: Student Edition" OUP, c1996
- [14] Rorabaugh, C. Britton, "Error Coding Cookbook" McGraw Hill, c1996
- [15] J. Romkey, "Non norme pour la [transmission des datagrammes IP](#) sur des lignes en série : SLIP", STD 47, RFC1055, juin 1988.
- [16] Recommended Practice for Line 21 Data Service (ANSI/EIA-608-94) (septembre 1994)
- [17] Stevens, W. Richard, "TCP/IP Illustrated, Volume 1,: The Protocols" Reading: Addison-Wesley Publishing Company, c1994.

10. Acronymes

FEC (*Forward Error Correction*) : correction d'erreur directe

IP (*Internet Protocol*) : protocole Internet

NABTS (*North American Basic Teletext Standard*) : norme nord américaine de télétexte de base

NTSC (*National Television Standards Committee*) : comité national des normes de télévision

NTSC-J (*NTSC-Japanese*) : comité japonais des normes de télévision

PAL (*Phase Alternation Line*) : ligne à phases alternatives

RFC (*Request for Comments*) : demande de commentaires

SECAM : Séquentiel Couleur Avec Mémoire

SLIP (*Serial Line Internet Protocol*) : protocole Internet de ligne en séquence

TCP (*Transmission Control Protocol*) : protocole de contrôle de transmission

UDP (*User Datagram Protocol*) : protocole de datagramme d'utilisateur

VBI (*Vertical Blanking Interval*) : intervalle de suppression vertical

WST (*World System Teletext*) : système de télétexte mondial

11. Adresse et contact des éditeurs

Ruston Panabaker, co-editor
Microsoft
One Microsoft Way
Redmond, WA 98052
mél : rustonp@microsoft.com

Simon Wegerif, co-editor
Philips Semiconductors
811 E. Arques Avenue
M/S 52, P.O. Box 3409
Sunnyvale, CA 94088-3409
mél : Simon.Wegerif@sv.sc.philips.com

Dan Zigmond, WG Chair
WebTV Networks
One Microsoft Way
Redmond, WA 98052
mél : djz@corp.webtv.net

12. Appendice A : Spécification de la correction d'erreur directe

Cette FEC est optimisée pour les données portées dans le VBI d'un signal de télévision. Le télétexte a été utilisé pendant de nombreuses années et les erreurs de transmission de données ont été catégorisées en trois types principaux : 1) erreurs sur un seul bit à répartition aléatoire, 2) perte de lignes de données NABTS, 3) erreurs en salve.

La quantité et la distribution de ces erreurs est très variable et dépend de nombreux facteurs. La FEC est conçue pour réparer tous ces types d'erreurs.

12.1 Mathématiques de la FEC

Les champs de Galois forment la base de l'algorithme de FEC présenté ici. Plutôt que d'expliquer ces champs en général, on donne une explication spécifique du champ de Galois utilisé dans l'algorithme de FEC.

Le champ de Galois utilisé est $GF(2^8)$ avec un élément alpha primitif de 00011101. C'est un ensemble de 256 éléments, avec les opérations "addition", "soustraction", "division", et "multiplication" sur ces éléments. Un nombre binaire de huit bits représente chaque élément.

Les opérations "addition" et "soustraction" sont les mêmes pour ce champ de Galois. Les deux opérations sont le OUX de deux éléments. Donc, la "somme" de 00011011 et 00000111 est 00011100.

La division de deux éléments est faite en utilisant la longue division avec les opérations de soustraction remplacées par OUX. Pour la multiplication, la multiplication standard longue est utilisée mais avec l'étape d'addition finale remplacée par OUX.

Toute l'arithmétique de la FEC qui suit est faite modulo 100011101 ; par exemple, après avoir multiplié deux nombres, on remplace le résultat par son reste divisé par 100011101. Il y a 256 valeurs dans ce champ (256 restes possibles) les nombres de 8 bits. Il est très important de rappeler que lorsque on écrit $A*B = C$, on veut dire plus précisément modulo $(A*B) = C$.

Il est clair d'après la description ci-dessus que multiplication et division prennent du temps à effectuer. Les éléments du champ de Galois partagent deux importantes propriétés avec les nombres réels.

$$A*B = \text{PUISSANCEalpha}(\text{LOGalpha}(A) + \text{LOGalpha}(B))$$

$$A/B = \text{PUISSANCEalpha}(\text{LOGalpha}(A) - \text{LOGalpha}(B))$$

Le champ de Galois est limité à 256 entrées de sorte que les tableaux de puissances et de logarithmes sont limités à 256 entrées. L'addition et la soustraction montrées sont standard de sorte que le résultat doit être modulo alpha. Elles sont écrites comme une expression "C" :

$$A*B = \text{apuissance}[\text{alog}[A] + \text{alog}[B]]$$

$$A/B = \text{apuissance}[255 + \text{alog}[A] - \text{alog}[B]]$$

On note que $\text{alog}[A] + \text{alog}[B]$ peut être supérieur à 255 et qu'une opération modulo devrait donc être effectuée. Cela n'est pas nécessaire si le tableau des puissances est étendu à 512 entrées par répétition du tableau. La même logique est vraie pour la division. Les tableaux de puissances et de logarithmes sont calculés à l'aide de la multiplication longue montrée plus haut.

12.2 Calcul des octets de FEC

L'algorithme de FEC partage un flux séquentiel en "ballots". Ceux-ci sont arrangés en 16 paquets de 28 octets lorsque les octets de correction sont inclus. Le ballot a donc 16 codets horizontaux entrelacés avec 28 codets verticaux. Deux sommes sont calculées pour un codet, S0 et S1. S0 est la somme de tous les octets du codet, chacun multiplié par alpha à la puissance de son indice dans le codet. S1 est la somme de tous les octets du codet, chacun multiplié par alpha à la puissance de trois fois son indice dans le codet. En "C" les calculs de somme ressembleraient à :

```
Somme0 = 0;
Somme1 = 0;
Pour (i = 0; i < m; i++)
{
  si (codet[i] != 0)
  {
    Somme0 = somme0 ^ puissance[i + alog[codet[i]]];
    Somme1 = somme1 ^ puissance[3*i + alog[codet[i]]];
  }
}
```

Noter que l'ordre des codets est différent de l'ordre des paquets. Les positions de codets 0 et 1 sont les octets de suffixe à la fin d'un paquet horizontalement ou à la fin d'une colonne verticalement.

Lors du calcul des deux octets de FEC, La sommation ci-dessus doit produire deux sommes de zéro. Tous les codets sauf 0 et 1 sont connus de sorte que les sommes pour les codets connus peuvent être calculées. Appelons ces valeurs tot0 et tot1.

$$\text{Somme0} = \text{tot0}^{\text{puissance}[0 + \text{alog}[\text{codet}[0]]]} \times \text{puissance}[1 + \text{alog}[\text{codet}[1]]]$$

$$\text{Somme1} = \text{tot1}^{\text{puissance}[0 + \text{alog}[\text{codet}[0]]]} \times \text{puissance}[3 + \text{alog}[\text{codet}[1]]]$$

Cela nous donne deux équations à deux inconnues qu'on peut résoudre :

$$\text{codet}[1] = \text{puissance}[255 + \text{alog}[\text{tot0}^{\text{tot1}}] - \text{alog}[\text{puissance}[1]^{\text{puissance}[3]}]]$$

$$\text{codet}[0] = \text{tot0}^{\text{puissance}[\text{alog}[\text{codet}[1]] + \text{alog}[\text{puissance}[1]]]}$$

12.3 Correction des erreurs

Ce paragraphe décrit la procédure de détection et correction des erreurs en utilisant les données de FEC calculées ci-dessus. À réception, on commence par reconstruire le ballot. C'est peut-être la partie la plus importante de la procédure parce que si le ballot n'est pas construit correctement, il ne peut pas corriger d'erreurs. L'indice de continuité est utilisé pour déterminer l'ordre des paquets et si des paquets sont manquants (non capturés par le matériel). La recommandation, lors de la construction du ballot, est de convertir le ballot de l'ordre des paquets en ordre des codets. Cette conversion va simplifier le calcul des codets. Cela est fait en prenant le dernier octet d'un paquet et en en faisant le second octet du codet et en prenant le second dernier octet d'un paquet et en en faisant le premier octet du codet. Ainsi, le paquet avec l'indice de continuité 15 devient la position de codet un et le paquet avec l'indice de continuité 14 devient la position de codet zéro. La même FEC est utilisée sans considération du nombre d'octet(s) dans le codet. Voyons donc les données comme une matrice $\text{codet}[\text{vert}][\text{hor}]$ où vert fait 16 paquets et hor en fait 28. Chaque octet dans la matrice est protégé par un codet horizontal et un codet vertical. Pour chacun des codets, les sommes doivent être calculées. Si les deux sommes pour un codet sont zéro, aucune erreur n'a été détectée pour ce codet. Autrement, une erreur a été détectée et d'autres étapes doivent être franchies pour voir si l'erreur peut être corrigée. En "C", la sommation horizontale ressemblerait à :

```

pour (i = 0; i < 16; i++)
{
  somme0 = 0;
  somme1 = 0;
  pour (j = 0; j < hor; j++)
  {
    si (codet[i][j] != 0)
    {
      somme0 = somme0 ^ puissance[j + alog[codet[i][j]]];
      somme1 = somme1 ^ puissance[3*j + alog[codet[i][j]]];
    }
  }
  si ((somme0 != 0) || (somme1 != 0))
  {
    Essayer de corriger le paquet
  }
}

```

De façon similaire, la sommation verticale ressemble à :

```

pour (j = 0; j < hor; j++)
{
  somme0 = 0;
  somme1 = 0;
  pour (i = 0; i < 16; i++)
  {
    si (codet[i][j] != 0)
    {
      somme0 = somme0 ^ puissance[i + alog[codet[i][j]]];
      somme1 = somme1 ^ puissance[3*i + alog[codet[i][j]]];
    }
  }
  si ((somme0 != 0) || (somme1 != 0))
  {
    Essayer de corriger la colonne
  }
}

```

12.4 Schémas de correction

Cette FEC donne quatre possibilités de correction :

- 1) Correction d'un seul bit dans le codet. Que des erreurs sur un seul bit.
- 2) Correction d'un double bit dans le codet. Surtout des erreurs sur deux bits.
- 3) Correction d'un seul octet dans un codet. Toutes les erreurs sur un seul octet.
- 4) Remplacement de paquet. Un ou deux paquets manquants dans un ballot.

12.4.1 Correction d'un seul bit

Lorsque on corrige un seul bit dans un codet, la position de l'octet et du bit doivent être calculées. les équations sont :

$$\begin{aligned} \text{Octet} &= 1/2 \text{LOGalpha}(S1/S0) \\ \text{Bit} &= 8 \text{LOGalpha}(S0/\text{PUISSANCEalpha}(\text{Byte})) \end{aligned}$$

En "C" cela s'écrit :

```
Octet = alog[puissance[255 + alog[somme1] - alog[somme0]]];
si (Octet & 1)
  Octet = Octet + 255;
Octet = Octet >> 1;
Bit = alog[puissance[255 + alog[somme0] - Octet]] << 3;
alors (Bit > 255)
  Bit = Bit - 255;
```

L'erreur est corrigible si Octet est inférieur au nombre d'octets dans le codet et si Bit est inférieur à huit. Pour que ce calcul soit valide somme0 et somme1 doivent toutes deux être différentes de zéro. Le codet est corrigé par :

$$\text{codet}[\text{Octet}] = \text{codet}[\text{Byte}] ^ (1 \ll \text{Bit});$$

12.4.2 Correction d'un double bit

La correction d'un double bit est beaucoup plus complexe que celle d'un seul bit pour deux raisons. D'abord, toutes les erreurs de double ne sont pas déterministes. C'est que deux différents schémas binaires peuvent générer les mêmes sommes. Ensuite, la solution est itérative. Pour trouver deux erreurs de bit on suppose qu'un bit est erroné et ensuite on résout pour la seconde erreur comme une erreur sur un seul bit.

La procédure est de se déplacer itérativement à travers les bits du codet en changeant l'état de chaque bit. Les nouvelles sommes sont calculées pour le codet modifié. Le calcul sur un seul bit ci-dessus détermine si c'est la solution correcte. Sinon, le bit est restauré et le bit suivant est essayé.

Pour un codet long, cela peut impliquer beaucoup de calculs. Cependant, des trucs peuvent accélérer le processus. Par exemple, les sommes verticales donnent une forte indication des octets qui sont erronés horizontalement. Les bits dans les autres octets n'ont pas besoin d'être essayés.

12.4.3 Correction d'un seul octet

Pour la correction d'un seul octet, la position des octets et des bits à corriger doit être calculée. Les équations sont :

$$\begin{aligned} \text{Octet} &= 1/2 * \text{LOGalpha}(S1/S0) \\ \text{Gabarit} &= S0/\text{PUISSANCEalpha}[\text{Octet}] \end{aligned}$$

On remarque que la position d'octet est le même calcul que pour une correction d'un seul bit. Le gabarit va permettre que soit corrigé plus d'un bit dans l'octet. En langage "C", le calcul du gabarit ressemble à :

$$\text{Gabarit} = \text{puissance}[255 + \text{alog}[\text{somme0}] - \text{Octet}]$$

somme0 et somme1 doivent toutes deux être différentes de zéro pour que les calculs soient valides. La valeur de Octet doit être inférieure à la longueur du codet mais Gabarit peut avoir n'importe quelle valeur. Cela corrige l'octet dans le codet avec :

$$\text{Codet}[\text{Octet}] = \text{Codet}[\text{Octet}] ^ \text{Gabarit}$$

12.4.4 Remplacement de paquet

Si un paquet est manquant, comme déterminé par l'indice de continuité, alors, sa position d'octet est connue et n'a pas besoin d'être calculée. La formule pour le remplacement d'un seul paquet est donc la même que pour le calcul du gabarit d'une correction d'un seul octet. Au lieu de OUXer un octet existant avec le gabarit, le gabarit remplace la position du codet manquant:

$$\text{Codet}[\text{Octet}] = \text{Gabarit}$$

Lorsque deux paquets manquent, les deux positions de codet sont connues par l'indice de continuité. Cela donne encore deux équations avec deux inconnues, qui sont résolues pour donner les équations suivantes.

$$\text{Gabarit2} = \text{PUISSANCEalpha}(2 * \text{Octet1}) * \text{S0} + \text{S1}$$

$$\text{PUISSANCEalpha}(2 * \text{Octet1} + \text{Octet2}) + \text{PUISSANCEalpha}(3 * \text{OCTET2})$$

$$\text{Gabarit1} = \text{S0} + \text{Gabarit2} * \text{PUISSANCEalpha}(\text{Octet2}) / \text{PUISSANCEalpha}(\text{OCTET1})$$

En "C", ces équations s'écrivent :

```

si (somme0 == 0)
{
  si (somme1 == 0)
    gabarit2 = 0;
  autrement
    gabarit2=puissance[255+alog[somme1]-alog[puissance[octet2+2*octet1]^puissance[3*Octet2]]];
}
autrement
{
  si ((a=somme1^puissance[alog[somme0]+2*octet1]) == 0)+gabarit2 = 0;
  autrement
    gabarit2 =puissance[255+alog[a]-alog[puissance[octet2+2*octet1]^puissance[3*octet2]]];
}

si (gabarit2 = 0)
{
  si (somme0 == 0)
    gabarit1 = 0;
  autrement
    gabarit1 = puissance[255+alog[somme0]-octet1];
}
autrement
{
  si ((a=somme0^puissance[alog[gabarit2] + octet2]) == 0)
    gabarit1 = 0;
  autrement
    gabarit1 = puissance[255+alog[a]-octet1];
}

```

Noter que dans le code ci-dessus, il faut veiller à vérifier les valeurs à zéro. La position du codet manquant peut être réparée par :

```

codet[octet1] = gabarit1;
codet[octet2] = gabarit2;

```

12.5 Considérations sur les performances de FEC

Le paragraphe précédent montre comment corriger les différents types d'erreurs. Il ne suggère pas comment ces corrections peuvent être utilisées dans un algorithme pour corriger un ballot. Il y a de nombreux algorithmes possibles et celui qui sera choisi dépend de nombreuses variables. Cela inclut :

- . la quantité de puissance de traitement disponible,
- . le nombre de paquets par VBI à traiter,

- . le type de matériel de saisie des données,
- . le chemin de livraison du VBI,
- . comment le code est mis en œuvre.

Au minimum, il est recommandé que l'algorithme utilise une correction d'un seul bit ou d'un seul octet pour une passe dans chaque direction, suivie par le remplacement du paquet si approprié. Il est possible de faire plus d'une passe de correction d'erreur dans chaque direction. La théorie dit que les erreurs non corrigées à la première passe peuvent être corrigées à la seconde passe parce que la correction d'erreur dans l'autre direction a retiré certaines erreurs.

Pour faire un choix, il est important de se souvenir que le code a plusieurs états possibles :

- 1) Montre le codet comme correct et il l'est.
- 2) Montre le codet comme correct et il ne l'est pas (échec de détection).
- 3) Montre le codet comme incorrect mais ne peut pas le corriger (détection).
- 4) Montre le codet comme incorrect et le corrige correctement (correction).
- 5) Montre le codet comme incorrect mais corrige les mauvais bits (fausse correction).

IL y a en fait des chevauchement entre les différents types d'erreurs. Par exemple, une paire de sommes peut indiquer à la fois une double erreur de bits et une erreur d'octet. Il n'est pas possible de savoir au niveau du code ce qui est une correction correcte et ce qui est une fausse correction. En fait, aucune peut n'être correcte si toutes deux sont de fausses corrections.

Si on sait quelque chose sur les types d'erreurs dans le canal de livraison, on peut améliorer grandement l'efficacité. Si on sait que les erreurs sont à distribution aléatoire (comme dans une diffusion terrestre faible) la correction de bit seul et double est plus puissante que celle d'un seul octet.

13. Appendice B : Architecture

L'architecture que vise le présent document peut être divisée en trois domaines : insertion, réseau de distribution, et client receveur.

L'insertion des données IP dans le signal de télévision peut survenir dans toute partie du système de livraison. Un codeur VBI accepte normalement un signal vidéo et un flux de série asynchrone d'octets formant des trames de paquets IP comme entrées et met ensuite les données en paquets sur un ensemble choisi de lignes en utilisant NABTS et une FEC. Ce signal composite est alors modulé avec d'autres canaux avant d'être diffusé sur le réseau de distribution. Les opérateurs qui sont plus loin dans la chaîne de distribution pourraient alors aussi ajouter leurs propres données, sur les autres lignes inutilisées. Le réseau de distribution inclut des installations en coaxial, par ondes hertziennes, et par systèmes de satellite analogique, et sont principalement des réseaux de diffusion unidirectionnels. Ils doivent fournir un ratio signal sur bruit, qui soit suffisant pour que la FEC récupère toutes données perdues pour que la diffusion des données soit efficace.

Le client receveur doit être capable de régler, d'échantillonner l'onde NABTS comme approprié, de filtrer de façon approprié les adresses NABTS, de faire la correction d'erreur directe, de déramer, de vérifier le CRC et de décompresser l'en-tête UDP/IP si il est compressé. Toutes ces fonctions peuvent être assurées par un logiciel de PC et un matériel peu coûteux du grand commerce.

14. Appendice C : Portée des protocoles proposés

Les protocoles décrits dans le présent document sont pour la transmission des paquets IP. Cependant, leur portée peut être extensible à d'autres applications en dehors de ce domaine. Beaucoup de protocoles du présent document pourraient être mis en œuvre sur n'importe quel réseau unidirectionnel.

Le protocole de tramage unidirectionnel fournit l'encapsulation de datagrammes IP sur un flux en série, et la compression des en-têtes UDP/IP réduit la redondance de transmission, préservant ainsi la bande passante. Ces deux protocoles pourraient être largement utilisés sur différents réseaux de diffusion unidirectionnels ou schémas de modulation pour transporter efficacement tout type de données en paquets. En particulier, de nouvelles versions du protocole Internet peuvent être prises en charge pour fournir une méthode normalisée de transport des données.

15. 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 définies dans les procédures des normes d'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.