

Groupe de travail Réseau  
**Request for Comments : 3150**  
**BCP 48**  
 Catégorie : Bonnes pratiques actuelles  
 Traduction Claude Brière de L'Isle

S. Dawkins  
 G. Montenegro  
 M. Kojo  
 V. Magret  
 juillet 2001

## Implications des liaisons lentes sur les performances de bout en bout

### Statut de ce mémoire

Le présent document spécifie les bonnes pratiques actuelles de l'Internet pour la communauté de l'Internet et appelle à des discussions et suggestions pour son amélioration. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de copyright

Copyright (C) The Internet Society (2001). Tous droits réservés.

### Résumé

Le présent document fait des recommandations en relation avec les performances pour les utilisateurs de chemins de réseau qui traversent des liaisons à "débit binaire très faible".

"Débit binaire très faible" implique "plus faible que ce qu'on voudrait". Cette recommandation peut être utile dans tout réseau où les hôtes peuvent saturer la bande passante disponible, mais l'espace de concept de cette recommandation inclut explicitement les connexions qui traversent des liaisons modem à 56 kbit/s ou des liaisons d'accès sans fil à 4,8 kbit/s – qui sont toutes deux largement déployées.

Le présent document expose les mécanismes d'utilité générale. Lorsque des mécanismes spécifiques de l'application peuvent faire mieux que les mécanismes d'utilisation générale pertinents, on le mentionne et on explique pourquoi.

Le présent document partage certaines recommandations avec la RFC2689, "Fournir des services intégrés sur des liaisons à faible débit binaire", en particulier dans les domaines comme la compression d'en-tête. Le présent document se concentre plus sur les applications de données traditionnelles pour lesquelles la "livraison au mieux" est appropriée.

## Table des Matières

|                                                                        |    |
|------------------------------------------------------------------------|----|
| 1. Introduction.....                                                   | 1  |
| 2. Description des optimisations.....                                  | 2  |
| 2.1 Autres solutions de compression d'en-tête.....                     | 2  |
| 2.2 Autres solutions de compression de charge utile.....               | 3  |
| 2.3 Choisir les tailles de MTU.....                                    | 3  |
| 2.4 Interactions avec le contrôle d'encombrement de TCP [RFC2581]..... | 4  |
| 2.5 Auto-réglage de mémoire tampon TCP.....                            | 5  |
| 2.6 Effets des petites fenêtres.....                                   | 6  |
| 3. Sommaire des optimisations recommandées.....                        | 6  |
| 4. Sujets pour des travaux futurs.....                                 | 7  |
| 5. Considérations pour la sécurité.....                                | 8  |
| 6. Considérations relatives à l'IANA.....                              | 8  |
| 7. Remerciements.....                                                  | 8  |
| 8. Références.....                                                     | 8  |
| Déclaration complète de droits de reproduction.....                    | 10 |

## 1. Introduction

La pile de protocoles Internet a été conçue pour fonctionner dans une large gamme de vitesses de liaisons, et a satisfait à cet objectif de conception avec seulement un nombre limité d'améliorations (par exemple, l'utilisation de l'adaptation de fenêtre TCP, décrite dans "Extensions à TCP pour de bonnes performances" [RFC1323] pour les connexions à très forte bande passante).

Les protocoles antérieurs à la Toile mondiale tendaient à être soit des applications interactives envoyant très peu de données (par exemple, Telnet) soit des applications de transfert en vrac qui n'exigeaient pas de réponse interactive (par exemple, le

protocole de transfert de fichiers, Nouvelles du réseau). La Toile mondiale nous a donné du trafic qui est à la fois interactif et souvent "en vrac", incluant des images, du son, et de la vidéo.

La Toile mondiale a aussi popularisé l'Internet, de sorte qu'il y a un intérêt significatif à accéder à l'Internet à des vitesses de liaisons qui sont beaucoup plus "lentes" que les vitesses normales de réseau d'affaires. En fait, une proportion significative des utilisateurs courants de l'Internet est connectée à l'Internet sur des liaisons de dernier bond relativement lentes. À l'avenir, le nombre de ces usagers va vraisemblablement augmenter rapidement car divers appareils mobiles sont prévus pour être rattachés à l'Internet sur des liaisons sans fils très lentes.

Afin de fournir la meilleure réponse interactive à ces transferts "en vrac", les mises en œuvre peuvent souhaiter minimiser le nombre de bits effectivement transmis sur ces connexions "lentes". Ce sont deux domaines qui peuvent être considérés – compresser les bits qui constituent la redondance associée à la connexion, et compresser les bits qui constituent la charge utile transportée sur la connexion.

De plus, les mises en œuvre peuvent souhaiter considérer les mécanismes de réglage et de mise en file d'attente de fenêtre de réception TCP comme des techniques pour améliorer les performances sur les liaisons à basse vitesse. Bien que ces techniques n'impliquent pas de changement aux protocoles, elles sont incluses dans le présent document pour qu'il soit complet.

## 2. Description des optimisations

Cette section décrit les optimisations dont l'utilisation a été suggérée dans des situations où les hôtes peuvent saturer leurs liaisons. La section suivante résume les recommandations sur l'utilisation de ces optimisations.

### 2.1 Autres solutions de compression d'en-tête

Les mécanismes pour la compression d'en-tête TCP et IP définis dans les [RFC1144], [RFC2507], [RFC2508], [RFC2509], [RFC3095] procurent les avantages suivants :

- Amélioration du temps de réponse interactive
- Diminution de la redondance d'en-tête (pour une MTU normale d'appel numéroté de 296 octets, la redondance des en-têtes TCP/IP peut diminuer d'environ 13 pour cent avec des en-têtes normaux de 40 octets à 1-1,5 pour cent avec des en-têtes compressés de 3-5 octets, pour la plupart des paquets). Cela permet d'utiliser de petits paquets pour le trafic à faible débit de données sensible aux retards et une bonne efficacité de ligne pour les données en vrac même avec de petites tailles de segment (pour les raisons d'utiliser une petite MTU sur les liaisons lentes, voir le paragraphe 2.3).
- De nombreuses liaisons lentes sont aujourd'hui sans fils et tendent à subir des pertes significatives. La compression d'en-tête réduit le taux de pertes de paquet sur les lignes à pertes (simplement parce que de plus courts temps de transmission exposent les paquets à moins d'événements qui causent les pertes).

La compression d'en-tête de la [RFC1144] est une proposition de norme pour la compression d'en-tête TCP qui est largement répandue. Malheureusement, elle est vulnérable aux liaisons à pertes, parce que même un seul bit d'erreur résulte en une perte de synchronisation entre le compresseur et le décompresseur. Elle utilise les temporisations de TCP pour détecter une telle perte de synchronisation, mais ces erreurs résultent en pertes de données (jusqu'à une pleine fenêtre TCP) en retards d'un RTO complet, et en démarrages lents inutiles.

Une proposition plus récente de compression d'en-tête [RFC2507] comporte une demande explicite de retransmission d'un paquet non compressé pour permettre la resynchronisation sans attendre une temporisation TCP (et l'exécution des procédures d'évitement d'encombrement). Cela fonctionne beaucoup mieux sur les liaisons avec des caractéristiques de pertes.

Le schéma ci-dessus cesse de bien fonctionner dans des conditions aussi extrêmes que celles de nombreuses liaisons cellulaires (conditions d'erreur de  $1e-3$  ou  $1e-2$  et temps d'aller-retour supérieurs à 100 ms.). Pour ces cas, le groupe de travail 'Compression d'en-tête robuste' a développé ROHC [RFC3095]. Les extensions de ROHC qui prennent en charge la compression des en-têtes TCP sont aussi en cours de développement.

La [RFC1323] définit une option "Horodatages TCP", utilisée pour empêcher le "retour à zéro" de l'espace de numéros de séquence TCP sur les liaisons à grande vitesse, et pour améliorer les estimations de RTT de TCP en fournissant des mesures non ambiguës de délais d'aller-retour TCP. L'utilisation des horodatages TCP empêche la compression d'en-tête, parce que les horodatages sont envoyés comme des options TCP. Cela signifie que chaque en-tête horodaté a des options TCP qui diffèrent de l'en-tête précédent, et les en-têtes qui changent d'options TCP sont toujours envoyés non compressés. De plus, les horodatages ne semblent pas avoir beaucoup d'impact sur l'estimation du RTO [AIPa99].

Néanmoins, le groupe de travail ROHC développe des schémas pour compresser les en-têtes TCP, incluant des options comme les horodatages et les accusées de réception sélectifs.

Recommandation : Mettre en œuvre la [RFC2507], en particulier en ce qui concerne les tunnels IPv4 et l'encapsulation minimale pour IP mobile, ainsi que la compression d'en-tête TCP pour les liaisons à pertes et les liaisons qui réordonnent les paquets. Les appareils à capacité PPP devraient mettre en œuvre la "compression d'en-tête IP sur PPP" [RFC2509]. La compression d'en-tête robuste [RFC3095] est recommandée pour les liaisons extrêmement lentes avec de très forts taux d'erreurs (voir ci-dessus) mais les mises en œuvre devraient juger si sa complexité est justifiée (peut-être par le coût des ressources radio fréquences).

La compression d'en-tête de la [RFC1144] ne devrait être activée que sur des liaisons "lentes" fiables.

L'utilisation des horodatages TCP [RFC1323] n'est pas recommandée avec ces connexions, parce qu'elle complique la compression d'en-tête. Même si le groupe de travail Compression d'en-tête robuste (ROHC) développe des spécifications pour y remédier, ces mécanismes ne sont pas encore pleinement développés ni déployés, et ne peuvent pas être généralement justifiables. De plus, les connexions qui traversent des liaisons "lentes" n'exigent pas de protection contre le retour à zéro de numéro de séquence TCP.

## 2.2 Autres solutions de compression de charge utile

La compression des charges utiles IP est aussi souhaitable sur les liaisons de réseau "lentes". La [RFC2393] "Protocole de compression de charge utile IP (IPComp)" définit un cadre où des algorithmes de compression courants peuvent être appliqués à des charges utiles de segments IP arbitraires.

La compression de charge utile IP est en quelque sorte une optimisation de niche. Elle est nécessaire parce que la sécurité de niveau IP convertit les charges utiles IP en flux binaires arbitraires, déjouant les mécanismes de compression de couche liaison couramment déployés qui se trouvent en face de charges utiles qui n'ont pas "d'informations" redondantes qui puissent être représentées de façon plus compacte.

Cependant, de nombreuses charges utiles IP sont déjà compressées (fichiers images, audio, vidéo, "zippés" qu'on transfère) ou sont déjà chiffrés au dessus de la couche IP (par exemple, SSL [SSL]/TLS [RFC2246]). Ces charges utiles ne seront pas plus "compressées", limitant les bénéfices de cette optimisation.

Pour les types de charge utile HTTP non compressés, HTTP/1.1 [RFC2616] inclut aussi les en-têtes Content-Encoding et Accept-Encoding, prenant en charge divers algorithmes de compression pour les types MIME compressibles courants comme du texte en clair. Cela ne laisse seulement non compressés que les en-têtes HTTP eux-mêmes.

En général, la compression de niveau application peut souvent faire mieux que IPComp, parce que l'opportunité d'utiliser les dictionnaires de compression fondés sur la connaissance des données spécifiques à compresser.

Une utilisation extensive des techniques de compression de niveau application va diminuer le besoin d'IPComp, en particulier pour les utilisateurs de la Toile mondiale.

Recommandation : IPComp peut facultativement être mis en œuvre.

## 2.3 Choisir les tailles de MTU

Plusieurs points sont à garder en mémoire lors du choix d'une MTU pour les liaisons à faible vitesse.

D'abord, si une MTU de pleine longueur occupe une liaison pour plus longtemps que la temporisation d'un accusé de réception (ACK) retardé (normalement 200 millisecondes, mais peut aller jusqu'à 500 millisecondes) cette temporisation va causer la génération d'un ACK pour chaque segment, plutôt que pour chaque second segment, comme cela arrive avec la plupart des mises en œuvre d'algorithme d'ACK retardé de TCP.

Ensuite, les MTU "relativement grandes", qui prennent un temps perceptible par l'homme pour être transmises dans le réseau, créent des délais perceptibles par l'homme dans d'autres flux utilisant la même liaison. La [RFC1144] considère des délais de 100 à 200 millisecondes comme perceptibles par l'homme. La convention de choisir des MTU de 296 octets (avec compression d'en-tête activée) pour l'accès téléphonique est un compromis qui limite le délai maximum d'occupation de liaison avec des MTU de pleine longueur proche de 200 millisecondes sur des liaisons à 9,6 kbit/s.

Troisièmement, sur les liaisons de dernier bond, utiliser une plus grande taille de MTU, et donc une plus grande taille de segment maximum (MSS, *Maximum Segment Size*), permettrait à un expéditeur TCP d'augmenter sa fenêtre d'encombrement en octets plus vite que lorsque il utilise une plus petite taille de MTU (et une plus petite MSS). Cependant, avec une plus petite taille de MTU, et une plus petite taille de MSS, la fenêtre d'encombrement, lorsque mesurée en segments, augmente plus vite qu'elle ne ferait avec une plus grande taille de MSS. Les connexions qui utilisent de plus petites tailles de MSS seront plus vraisemblablement capables d'envoyer assez de segments pour générer trois accusés de réception dupliqués, déclenchant la retransmission/récupération rapide lorsque les pertes de paquet se rencontrent. Et donc, une plus petite taille de MTU est utile pour les liaisons lentes avec des caractéristiques de perte.

Quatrièmement, utiliser une plus petite taille de MTU diminue aussi le délai de mise en file d'attente d'un flux TCP (et donc le RTT) comparé à l'utilisation de plus grandes tailles de MTU avec le même nombre de paquets dans une file d'attente. Cela signifie qu'un flux TCP qui utilise une plus petite taille de segment et qui traverse une liaison lente est capable de porter la fenêtre d'encombrement (en nombre de segments) à une plus grande valeur tout en subissant le même délai de mise en file d'attente.

Finalement, certains réseaux tarifient le trafic sur la base du paquet, non sur celle du kilo octet. Dans ces cas, les connexions qui utilisent une plus grande MTU peut payer moins que des connexions qui transfèrent le même nombre d'octets en utilisant une plus petite MTU.

Recommandation : Si il est possible de le faire, les MTU devraient être choisies comme ne monopolisant pas les interfaces réseau pendant des durées perceptibles par l'homme, et les mises en œuvre ne devraient pas choisir des MTU qui vont occuper une interface réseau pour un temps significativement supérieur à 100-200 millisecondes.

## 2.4 Interactions avec le contrôle d'encombrement de TCP [RFC2581]

Dans de nombreux cas, les connexions TCP qui traversent une liaison lente ont celle-ci comme liaison "d'accès", avec des lignes à plus fort débit pour la plus grande partie du chemin de connexion. Une configuration courante peut être celle d'un ordinateur portable qui utilise un accès téléphonique à un serveur terminal (un routeur de dernier bond) avec un serveur HTTP sur un LAN à grande vitesse "derrière" le serveur terminal.

Dans ce cas, le serveur HTTP peut être capable de placer des paquets sur son LAN à grande vitesse directement rattaché au plus fort débit que le routeur de dernier bond peut leur assurer sur la liaison à basse vitesse. Lorsque le routeur de dernier bond tombe derrière, il n'est pas capable de mettre en mémoire tampon le trafic destiné à la liaison à basse vitesse, et va devenir un point d'encombrement et commencer à éliminer les paquets en excès. En particulier, plusieurs paquets peuvent être abandonnés dans une seule fenêtre de transmission lorsque le démarrage lent initial déborde de la mémoire tampon du routeur de dernier bond.

Bien que surviennent des pertes de paquets, elles ne sont pas détectées chez l'expéditeur TCP avant une durée de RTT après l'épuisement de l'espace de mémoire tampon du routeur et l'abandon du premier paquet. Ce signal d'encombrement tardif permet à la fenêtre d'encombrement d'augmenter jusqu'au double de la taille qu'elle avait au moment où le premier paquet a été abandonné au routeur.

Si la MTU de la liaison est assez grande pour prendre plus que l'intervalle de temporisation d'ACK retardé pour transmettre un paquet, un ACK est envoyé pour chaque segment et la fenêtre d'encombrement est doublée en un seul RTT. Si une plus petite MTU de liaison est utilisée et si des ACK retardés peuvent être utilisés, la fenêtre d'encombrement augmente d'un facteur 1,5 en un RTT. Dans les deux cas, l'expéditeur continue de transmettre des paquets bien au delà du point d'encombrement du routeur de dernier bond, ce qui résulte en plusieurs pertes de paquet dans une seule fenêtre.

La nature auto-rythmée des algorithmes TCP de démarrage lent et d'évitement d'encombrement empêche ce débordement de mémoire tampon de se poursuivre. De plus, ces algorithmes permettent aux expéditeurs de "sonder" la bande passante disponible – faisant un cycle d'augmentation de taux de transmission jusqu'à ce que surviennent des pertes, suivi par une chute rigoureuse (50 pour cent) du taux de transmission. Cela se produit lorsque un hôte directement connecté à une liaison à faible débit offre une fenêtre annoncée qui est trop large pour la liaison à bas débit. Durant la phase d'évitement d'encombrement, l'hôte homologue continue de sonder la bande passante disponible, essayant de remplir la fenêtre annoncée, jusqu'à ce que des pertes de paquet surviennent.

Les mêmes problèmes peuvent aussi exister quand un hôte expéditeur est directement connecté à une liaison lente, car la plupart des liaisons lentes ont une mémoire tampon locale dans l'interface de liaison. Cette mémoire tampon d'interface de liaison est sujette à débordement exactement de la même façon que la mémoire tampon de routeur de dernier bond.

Lorsque est utilisé un routeur de dernier bond avec un petit nombre de mémoires tampon par liaisons sortantes, le premier débordement de mémoire tampon survient plus tôt qu'il ne l'aurait fait si le routeur avait eu un plus grand nombre de mémoires tampon. Par conséquent, avec un plus petit nombre de mémoires tampon les pertes de paquet périodiques surviennent moins fréquemment durant l'évitement d'encombrement, lorsque l'envoyeur sonde la bande passante disponible.

La plus importante responsabilité des mémoires tampon de routeur est d'absorber les salves. Trop peu de mémoires tampon (par exemple, seulement trois mémoires tampon par liaison sortante comme décrit dans la [RFC2416]) signifie que les routeurs vont saturer très facilement leurs réservoirs de mémoire tampon et n'auront que peu de chances d'absorber même une très petite salve. Lorsque un plus grand nombre de mémoires tampon de routeur sont allouées par liaison sortante, l'espace de mémoire tampon ne se sature pas aussi rapidement mais les mémoires tampon ont des chances de se remplir à cause du comportement par défaut de TCP. Un plus grand nombre de mémoires tampon de routeur conduit à de plus longs délais de mise en file d'attente et un plus long RTT.

Si les files d'attente de routeur se remplissent avant que l'encombrement soit signalé ou restent pleines pendant de longues périodes, cela va vraisemblablement résulter en un "verrouillage", où une seule connexion ou quelques connexions occupent l'espace de file d'attente du routeur, empêchant les autres connexions d'utiliser la liaison [RFC2309], en particulier lorsque une discipline de gestion de file d'attente avec abandon de la queue est utilisée.

Donc, il est essentiel d'avoir un nombre assez grand de mémoires tampon dans un routeur pour qu'il soit capable d'absorber les salves de données, mais de garder la file d'attente raisonnablement petite. Pour réaliser cela, il a été recommandé dans la [RFC2309] qu'un mécanisme de gestion active de file d'attente, comme la détection précoce aléatoire (RED, *Random Early Detection*) [RED93] devrait être mis en œuvre dans tous les routeurs de l'Internet, incluant les routeurs de dernier bond en présence d'une liaison lente. On devrait aussi noter que RED exige un nombre suffisant de mémoires tampon de routeur pour fonctionner correctement. De plus, les paramètres appropriés de RED sur un routeur de dernier bond connecté à une liaison lente vont probablement s'écarter de ceux recommandés par défaut.

Un mécanisme actif de gestion de file d'attente n'élimine pas les abandons de paquet mais plutôt les abandons de paquets à une étape précédente pour résoudre le problème de la file d'attente complète pour les flux qui répondent aux abandons de paquet comme signal d'encombrement. Les hôtes qui sont directement connectés à des liaisons à basse vitesse peuvent limiter la fenêtre de receveur qu'ils annoncent afin de diminuer ou éliminer le nombre d'abandons de paquet chez un routeur de dernier bond. Lorsque on fait ainsi, on devrait cependant faire attention à ce que la fenêtre annoncée soit assez grande pour permettre une pleine utilisation de la capacité de la liaison de dernier bond et pour permettre de déclencher la retransmission rapide, en présence d'une perte de paquet. Cette recommandation prend deux formes :

- Les systèmes d'exploitation modernes utilisent des mémoire tampon de réception TCP par défaut relativement grandes par rapport à ce qui était exigé pour utiliser pleinement la capacité de liaison des liaisons à basse vitesse. Les usagers devraient être capables de choisir la taille de fenêtre de réception par défaut utilisée – normalement un paramètre à l'échelle du système. (Ce "choix" peut être aussi simple qu'un "accès téléphonique/accès de LAN" sur une boîte de dialogue – cela s'accommoderait à de nombreux environnements sans exiger de réglage manuel par des ingénieurs réseau expérimentés.)
- Les développeurs d'applications ne devraient pas tenter de gérer manuellement la bande passante du réseau en utilisant les tailles de mémoire tampon des prises. C'est seulement dans de très rares circonstances qu'une application connaît réellement la bande passante et le délai d'un chemin et est capable de choisir une valeur convenablement faible (ou élevée) pour la taille de mémoire tampon de prise pour obtenir de bonnes performances du réseau.

Cette recommandation n'est pas une solution générale pour tout chemin de réseau qui pourrait impliquer une liaison lente. Cette recommandation est plutôt applicable dans des environnements où l'hôte "sait" qu'il est toujours connecté aux autres hôtes via "des liaisons lentes". Pour les hôtes qui peuvent se connecter aux autres hôtes sur diverses liaisons (par exemple, des ordinateurs portables à numérotation téléphonique avec des stations connectées à un LAN) l'auto-réglage de mémoire tampon pour la mémoire tampon de réception est une recommandation plus raisonnable, qui est exposée ci-dessous.

## 2.5 Auto-réglage de mémoire tampon TCP

[SMM98] reconnaît une tension entre le désir d'allouer de "grandes" mémoires tampon TCP, afin que les chemins de réseau soient pleinement utilisés, et le désir de limiter la quantité de mémoire dédiée aux mémoires tampon TCP, afin de prendre en charge efficacement un grand nombre de connexions à des hôtes sur des chemins de réseau qui peuvent varier de six ordres de grandeur.

La technique proposée est d'allouer de façon dynamique les mémoires tampon TCP, sur la base de la fenêtre actuelle

d'encombrement, plutôt que de tenter de pré allouer les mémoires tampon TCP sans aucune connaissance du chemin du réseau.

Cette proposition a pour résultat que les mémoires tampon de réception sont appropriées aux tailles de fenêtre utilisées, et que les mémoires tampon d'envoi sont assez grandes pour contenir deux fenêtres de segments, de sorte que SACK et la récupération rapide puissent récupérer des pertes sans forcer la connexion à utiliser de longues temporisations de retransmission.

Bien que la plus grande part de la motivation de cette proposition soit donnée du point de vue du serveur, les hôtes qui se connectent en utilisant plusieurs interfaces avec des vitesses de liaison présentant des différences marquées peuvent aussi trouver utiles ces types de techniques. C'est en particulier vrai avec les liaisons lentes, qui vont vraisemblablement dominer le RTT de bout en bout. Si l'hôte est connecté via seulement une seule interface de liaison lente à la fois, il est très facile d'ajuster (de façon dynamique) la fenêtre de réception (et donc la fenêtre annoncée) à une valeur appropriée pour la liaison lente de dernier bond avec les caractéristiques connues de bande passant et de délai.

Recommandation : Si un hôte est parfois connecté via une liaison lente mais si l'hôte est aussi connecté en utilisant d'autres interfaces avec des vitesses de liaison présentant des différences marquées, il peut utiliser l'auto-réglage de mémoire tampon de réception pour ajuster la fenêtre annoncée à une valeur appropriée.

## 2.6 Effets des petites fenêtres

Si une connexion TCP se stabilise avec une fenêtre d'encombrement de seulement quelques segments (comme on pourrait s'y attendre sur une liaison "lente") l'expéditeur n'envoie pas assez de segments pour générer trois accusés de réception dupliqués, déclenchant la retransmission/récupération rapide. Cela signifie qu'une temporisation de retransmission est nécessaire pour réparer la perte – laissant la connexion TCP à une fenêtre d'encombrement avec un seul segment.

[TCPB98] et [TCPP98] observent (dans des études d'ensembles de données de trace de réseau) qu'il est relativement courant que les temporisations de retransmission TCP surviennent même lorsque quelques accusés de réception dupliqués sont envoyés. Le défi est d'utiliser ces accusés de réception dupliqués pour déclencher la retransmission/récupération rapide sans injecter inutilement de trafic dans le réseau – et en particulier de ne pas injecter de trafic d'une façon qui pourrait créer de l'instabilité.

L'algorithme de "transmission limitée" de la [RFC3042] suggère d'envoyer un nouveau segment lorsque le premier et le second accusés de réception dupliqués sont reçus, afin que le receveur soit plus vraisemblablement capable de continuer de générer des accusés de réception dupliqués jusqu'à ce que le seuil de retransmission TCP soit atteint, déclenchant la retransmission et la récupération rapides. Lorsque la fenêtre d'encombrement est petite, c'est très utile pour aider la retransmission et la récupération rapides à récupérer d'une perte de paquet sans utiliser une temporisation de retransmission. On note qu'un maximum de deux nouveaux segments supplémentaires seront envoyés avant que le receveur envoie soit un nouvel accusé de réception qui avance la fenêtre, soit deux accusés de réception dupliqués supplémentaires, déclenchant la retransmission/récupération rapide, et que ces nouveaux segments seront rythmés par les accusés de réception, et non dos à dos.

Recommandation : La transmission limitée devrait être mise en œuvre dans tous les hôtes.

## 3. Sommaire des optimisations recommandées

Cette section résume nos recommandations en ce qui concerne les mécanismes de normalisation précédents pour les nœuds d'extrémité connectés via une liaison lente.

La compression d'en-tête devrait être mise en œuvre. La compression d'en-tête de la [RFC1144] peut être activée sur des liaisons réseau robustes. La [RFC2507] devrait être utilisée sur des connexions réseau dont on s'attend à ce qu'elles subissent des pertes dues à la corruption ainsi qu'à l'encombrement. Pour les lignes lentes et extrêmement enclines aux pertes, les mises en œuvre devraient évaluer ROHC [RFC3095] comme solution potentielle. Les horodatages TCP de la [RFC1323] doivent être désactivés parce que (1) leur protection contre le retour à zéro du numéro de séquence est injustifié pour les liaisons lentes, et (2) ils compliquent la compression d'en-tête TCP.

La compression de charge utile IP de la [RFC2393] devrait être mise en œuvre bien que la compression à des couches supérieures de la pile de protocole (par exemple de la [RFC 2616]) peut rendre ce mécanisme moins utile.

Pour les environnements HTTP/1.1, la compression de charge utile de la [RFC2616] devrait être mise en œuvre et devrait

être utilisée pour les charges utiles qui ne sont pas déjà compressées.

Les mises en œuvre devraient choisir des MTU qui ne monopolisent pas les interfaces réseau pendant plus de 100 à 200 millisecondes, afin de limiter l'impact d'une seule connexion sur toutes les autres qui partagent l'interface réseau.

L'utilisation d'une gestion active de file d'attente est recommandée sur les routeurs de dernier bond qui fournissent l'accès Internet aux hôtes derrière une liaison lente. De plus, le nombre de mémoires tampon de routeur par liaison lente devrait être assez grand pour absorber les salves de données concurrentes pour plus d'un seul flux. Pour absorber les salves de données concurrentes provenant de deux ou trois envoyeurs TCP avec une salve normale de données de trois segments dos à dos par envoyeur, au moins six (6) ou neuf (9) mémoires tampon sont nécessaires. L'utilisation effective d'une gestion de file d'attente active exigera même vraisemblablement un plus grand nombre de mémoires tampon.

Les mises en œuvre devraient examiner la possibilité qu'un hôte soit directement connecté à une liaison lente lors du choix des tailles de fenêtre de réception TCP par défaut.

Les développeurs d'applications ne devraient pas tenter de gérer manuellement la bande passante du réseau en utilisant les tailles de mémoire tampon des prises car c'est seulement dans de très rares circonstances qu'une application sera capable de choisir une valeur convenable pour la taille de mémoire tampon de la prise pour obtenir de bonnes performances du réseau.

La transmission limitée de la [RFC3042] devrait être mise en œuvre dans tous les hôtes d'extrémité car elle aide à déclencher la retransmission rapide lorsque la fenêtre d'encombrement est petite.

Tous les mécanismes décrits ci-dessus sont des RFC stables en cours de normalisation (au statut de proposition de norme au moment de la rédaction).

De plus, les mises en œuvre peuvent souhaiter prendre en considération l'auto-réglage de mémoire tampon TCP, en particulier lorsque le système hôte va vraisemblablement être utilisé avec une grande diversité de vitesses de liaison d'accès. Ce n'est pas un mécanisme TCP en cours de normalisation mais, comme c'est un problème de mise en œuvre du système d'exploitation, il n'a pas besoin d'être normalisé.

Des mécanismes ci-dessus, seule la compression d'en-tête (pour IP et TCP) peut cesser de fonctionner en présence d'IPsec de bout en bout. Cependant, la [RFC3095] permet la compression de l'en-tête ESP.

#### 4. Sujets pour des travaux futurs

En plus des mécanismes en cours de normalisation discutés ci-dessus, il y a encore des opportunités d'amélioration des performances sur les liaisons à basse vitesse.

"Envoyer moins de bits" est une réponse évidente aux basses vitesses de liaison. La maintenant défunte proposition de [HTTP-NG] remplaçait la représentation d'en-tête HTTP fondée sur le texte par une représentation binaire pour être plus compacte. Cependant, HTTP-NG n'avance plus et HTTP/1.1 ne va pas être amélioré pour inclure une représentation plus compacte des en-têtes HTTP. À la place, le Forum Protocole d'applications sans fils (WAP, *Wireless Application Protocol*) a opté pour le protocole de session sans fils fondé sur XML (XSP, *XML-based Wireless Session Protocol*) [WSP], qui comporte un mécanisme de codage d'en-tête compact.

Il serait agréable de se mettre d'accord sur une représentation d'en-tête plus compacte qui serait utilisée par toutes les communautés de la Toile mondiale, et non seulement par la communauté du WAN sans fils. Bien sûr, les codages généraux de contenu XML ont été proposés [Millau], bien qu'ils ne soient pas encore largement adoptés.

On note que les options TCP qui changent de segment à segment désactivent effectivement les schémas de compression d'en-tête déployés aujourd'hui, parce qu'il n'y a aucun moyen d'indiquer que certains champs de l'en-tête sont changés par rapport au précédent segment, alors que d'autres champs le sont. Le groupe de travail Compression d'en-tête robuste développe de tels schémas pour les options TCP comme les horodatages et les accusés de réception sélectifs. On peut espérer que des documents définiront à la suite de la [RFC3095] de telles spécifications.

Un autre effort qu'il vaut la peine de suivre est celui des 'codages de delta'. Ici, les clients qui demandent une version légèrement modifiée d'une ressource précédemment mise en antémémoire vont recevoir une description succincte des différences, plutôt que la ressource entière [RFC3229].

## 5. Considérations pour la sécurité

Toutes les recommandations incluses dans le présent document sont des RFC stables en cours de normalisation (au statut de normes proposées au moment de la rédaction) et on ne suggère par ailleurs aucun changement à des protocoles. À l'exception de la compression de Van Jacobson [RFC1144], [RFC2507], [RFC2508], et [RFC2509], tous les autres mécanismes sont applicables aux connexions TCP protégées de bout en bout par IPsec. Cela inclut ROHC [RFC3095], bien que partiellement, parce que même si il peut compresser dans une certaine mesure l'en-tête ESP le plus externe, le chiffrement rend quand même incompressibles toutes les données de charge utile (y compris tous les en-têtes de protocole suivants).

## 6. Considérations relatives à l'IANA

Le présent document est un pointeur sur d'autres normes existantes de l'IETF. Il n'y a pas de nouvelles considérations relatives à l'IANA.

## 7. Remerciements

Cette recommandation découle de "Longs réseaux fins" [RFC2757], qui a son tour a bénéficié des travaux du groupe de travail TCPSAT de l'IETF.

## 8. Références

- [AlPa99] Mark Allman et Vern Paxson, "On Estimating End-to-End Network Path Properties", dans ACM SIGCOMM 99 Proceedings, 1999.
- [HTTP-NG] Mike Spreitzer, Bill Janssen, "HTTP 'Next Generation'", 9th International WWW Conference, mai 2000. Aussi disponible à : <http://www.www9.org/w9cdrom/60/60.html>
- [Millau] Marc Girardot, Neel Sundaresan, "Millau: an encoding format for efficient representation and exchange of XML over the Web", 9th International WWW Conference, mai 2000. Disponible à : <http://www.www9.org/w9cdrom/154/154.html>
- [PAX97] Paxson, V., "End-to-End Internet Packet Dynamics", 1997, dans SIGCOMM 97 Proceedings. Disponible à : <http://www.acm.org/sigcomm/ccr/archive/ccr-toc/ccr-toc-97.html>
- [RED93] Floyd, S., et Jacobson, V., "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, V.1 N.4, août 1993, pp. 397-413. Disponible à <http://ftp.ee.lbl.gov/floyd/red.html>
- [RFC1144] V. Jacobson, "Compression des en-têtes TCP/IP pour les liaisons série à faible débit", février 1990.
- [RFC1323] V. Jacobson, R. Braden et D. Borman, "[Extensions TCP](#) pour de bonnes performances", mai 1992.
- [RFC2246] T. Dierks et C. Allen, "[Protocole TLS version 1.0](#)", janvier 1999.
- [RFC2309] B. Braden et autres, "Recommandations sur la [gestion de file d'attente et l'évitement d'encombrement](#) dans l'Internet", avril 1998.
- [RFC2393] A. Shacham et autres, "Protocole de compression de charge utile IP (IPComp)", décembre 1998. (*Obs., voir [RFC3173](#)*)
- [RFC2401] S. Kent et R. Atkinson, "[Architecture de sécurité](#) pour le protocole Internet", novembre 1998. (*Obsolète, voir [RFC4301](#)*)
- [RFC2416] T. Shepard, C. Partridge, "Lorsque TCP commence par quatre paquets dans seulement trois mémoires tampon", septembre 1998. (*Information*)
- [RFC2507] M. Degermark, B. Nordgren, S. Pink, "[Compression d'en-tête IP](#)", février 1999. (*P.S.*)

- [RFC2508] S. Casner, V. Jacobson, "Compression d'en-têtes IP/UDP/RTP pour liaisons séries à bas débit", février 1999.
- [RFC2509] M. Engan, S. Casner, C. Bormann, "[Compression d'en-tête IP](#) sur PPP", février 1999. (*Obsolète, voir RFC3544*) (P.S.)
- [RFC2581] M. Alman, V. Paxson et W. Stevens, "[Contrôle d'encombrement avec TCP](#)", avril 1999. (*Obsolète, voir RFC5681*)
- [RFC2616] R. Fielding et autres, "[Protocole de transfert hypertexte](#) -- HTTP/1.1", juin 1999. (*D.S., MàJ par 2817, 6585*)
- [RFC2757] G. Montenegro et autres, "Longs réseaux fins", janvier 2000. (*Information*)
- [RFC3042] M. Allman, H. Balakrishnan, S. Floyd, "[Amélioration de la récupération de perte](#) dans TCP avec la transmission limitée", janvier 2001. (P.S.)
- [RFC3095] C. Bormann et autres, "[Compression d'en-tête robuste](#) (ROHC) : cadre et quatre profils", juillet 2001. (*MàJ par RFC3759, RFC4815*) (P.S.)
- [RFC3229] J. Mogul et autres, "[Codage des delta dans HTTP](#)", janvier 2002. (P.S.)
- [SMM98] Jeffrey Semke, Matthew Mathis, et Jamshid Mahdavi, "Automatic TCP Buffer Tuning", dans ACM SIGCOMM 98 Proceedings 1998. Disponible à [http://www.acm.org/sigcomm/sigcomm98/tp/abs\\_26.html](http://www.acm.org/sigcomm/sigcomm98/tp/abs_26.html) .
- [SSL] Alan O. Freier, Philip Karlton, Paul C. Kocher, "The SSL Protocol: Version 3.0", mars 1996. (Projet Internet arrivé à expiration, disponible à <http://home.netscape.com/eng/ssl3/ssl-toc.html> )
- [TCPB98] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, Mark Stemm, Randy H. Katz, "TCP Behavior of a Busy Internet Server: Analysis and Improvements", IEEE Infocom, mars 1998. Disponible à : <http://www.cs.berkeley.edu/~hari/papers/infocom98.ps.gz>
- [TCPF98] Dong Lin et H.T. Kung, "TCP Fast Recovery Strategies: Analysis and Improvements", IEEE Infocom, mars 1998. Disponible à : <http://www.eecs.harvard.edu/networking/papers/infocom-tcp-final-198.pdf>
- [WSP] Wireless Application Protocol Forum, "WAP Wireless Session Protocol Specification", approuvé le 4 mai 2000, disponible à <http://www1.wapforum.org/tech/documents/WAP-203-WSP-20000504-a.pdf> (référence pour information).

#### Adresse des auteurs

Les questions sur le présent document peuvent être adressées à :

Spencer Dawkins  
Fujitsu Network Communications  
2801 Telecom Parkway  
Richardson, Texas 75082  
téléphone : +1-972-479-3782  
mél : [spencer.dawkins@fnc.fujitsu.com](mailto:spencer.dawkins@fnc.fujitsu.com)

Gabriel Montenegro  
Sun Microsystems Laboratories, Europe  
29, chemin du Vieux Chene  
38240 Meylan, FRANCE  
téléphone : +33 476 18 80 45  
mél : [gab@sun.com](mailto:gab@sun.com)

Markku Kojo  
Department of Computer Science  
University of Helsinki  
P.O. Box 26 (Teollisuuskatu 23)  
FIN-00014 HELSINKI  
Finland  
téléphone : +358-9-1914-4179  
mél : [kojo@cs.helsinki.fi](mailto:kojo@cs.helsinki.fi)

Vincent Magret  
Alcatel Internetworking, Inc.  
26801 W. Agoura road  
Calabasas, CA, 91301  
téléphone : +1 818 878 4485  
mél : [vincent.magret@alcatel.com](mailto:vincent.magret@alcatel.com)

## **Déclaration complète de droits de reproduction**

Copyright (C) The Internet Society (2001). Tous droits réservés.

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

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

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

### **Remerciement**

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