

Groupe de travail Réseau  
**Request for Comments : 3366**  
**BCP : 62**  
Catégorie : Bonnes pratiques actuelles

G. Fairhurst, University of Aberdeen  
L. Wood, Cisco Systems Ltd  
août 2002  
Traduction Claude Brière de L'Isle

## Conseils aux concepteurs de liaisons sur la répétition automatique de demande (ARQ)

### Statut de ce mémoire

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

### Notice de copyright

Copyright (C) The Internet Society (2002). Tous droits réservés.

### Résumé

Le présent document donne des conseils aux concepteurs d'équipements de communication numériques et de protocoles de couche liaison qui emploient des techniques de demande de répétition automatique (ARQ, *Automatic Repeat reQuest*) de couche liaison. Le présent document présuppose que les concepteurs souhaitent prendre en charge les protocoles Internet, mais pourraient n'être pas familiers de l'architecture de l'Internet et des implications de leurs choix de conception sur les performances et l'efficacité du trafic Internet porté sur leurs liaisons.

ARQ est décrit d'une façon générale qui inclut son utilisation sur une large gamme de supports physiques sous-jacents, incluant le cellulaire sans fil, les LAN sans fil, les liaisons radio fréquence et les autres types de canal. Le présent document décrit aussi les questions pertinentes pour la prise en charge du trafic IP sur des canaux de couche physique où les performances varient, et où l'ARQ de liaison va probablement être utilisé.

### Table des Matières

1. Introduction.....	2
1.1 ARQ de liaison.....	2
1.2 Causes des pertes de paquet sur une liaison.....	3
1.3 Pourquoi utiliser ARQ ?.....	4
1.4 Techniques d'ARQ couramment utilisées.....	4
1.5 Causes de délai sur une liaison.....	5
2. Persistance d'ARQ.....	6
2.1 Protocoles d'ARQ parfaitement persistants (fiabiles).....	6
2.2 Protocoles d'ARQ à haute persistance (très fiabiles).....	7
2.3 Protocoles d'ARQ à faible persistance (partiellement fiabiles).....	7
2.4 Choisir sa persistance.....	8
2.5 Impact sur les pannes de liaison.....	8
3. Traitement des paquets et des flux.....	9
3.1 Ordre des paquets.....	9
3.2 Utilisation de l'ARQ de liaison pour la prise en charge de flux multiples.....	10
3.3 Différenciation des classes de service de liaison.....	10
4. Conclusions.....	11
5. Considérations pour la sécurité.....	12
6. Considérations relatives à l'IANA.....	12
7. Remerciements.....	12
8. Références.....	13
8.1 Références normatives.....	13
8.2 Références pour information.....	13
Adresse des auteurs.....	15
Déclaration complète de droits de reproduction.....	15

## 1. Introduction

Le protocole Internet (IP, *Internet Protocol*) [RFC791], forme le protocole central de l'Internet mondial et définit un simple réseau à commutation de paquet "sans connexion". Au fil des ans, le trafic Internet qui utilise IP a été porté sur une grande variété de liaisons, de capacités, de délais et de caractéristiques de pertes très différentes. À l'avenir, on peut s'attendre à ce que le trafic IP continue d'être transporté sur des concepts de liaisons nouveaux et existants très divers, là encore avec des caractéristiques très diverses.

Un document voisin, la [RFC3819], décrit les questions générales associées à la conception de liaison. Le présent document devrait être lu en conjonction avec lui et avec les autres documents produits par le groupe de travail Implications de performances des caractéristiques des liaisons (PILC) de l'IETF [RFC3135], [RFC3155].

Le présent document est destiné à trois groupes distincts de lecteurs :

- a. Les concepteurs de liaisons qui souhaitent configurer (ou régler) une liaison pour le trafic IP qu'elle va transporter, en utilisant les mécanismes standard de couche de liaison tels que le contrôle de liaison de données de haut niveau (HDLC, *High-level Data Link Control*) de l'ISO, [ISO4335a] ou ses dérivés.
- b. Ceux qui mettent en œuvre des liaisons et souhaitent concevoir de nouveaux mécanismes de liaison qui se comportent bien avec le trafic IP.
- c. La communauté des gens qui utilisent ou développent TCP, UDP et les protocoles qui s'y rapportent, qui peuvent souhaiter être au courant de la façon dont ces liaisons peuvent fonctionner.

Les audiences principales sont destinées aux groupes (a) et (b). Le groupe (c) ne devrait pas avoir besoin de connaître les détails exacts d'un schéma d'ARQ à travers une seule liaison, et ne devrait pas avoir à considérer de tels détails pour les mises en œuvre de protocole ; une grande partie de l'Internet fonctionne à travers des liaisons qui n'utilisent aucune forme d'ARQ. Cependant, la communauté TCP/IP n'a pas besoin de connaître la façon dont le protocole IP fonctionne sur les diverses gammes de sous-réseaux sous-jacents. Le présent document peut aider à élever ce niveau de connaissance.

Une fiabilité parfaite n'est pas une exigence pour les réseaux IP, et ce n'est pas une exigence pour les liaisons [RFC3819]. Les réseaux IP peuvent éliminer des paquets pour des raisons diverses sans aucune relation avec des erreurs de canaux, incluant un manque d'espace de file d'attente, la gestion de l'encombrement, des fautes, et des changements de chemins. On a compris depuis longtemps que la fiabilité parfaite de bout en bout ne peut être assurée qu'à la couche transport [SALT81] ou au dessus.

On suppose ici une certaine familiarité avec TCP, le protocole de contrôle de transmission [RFC793], [STE94]. TCP fournit un service de transport fiable de flux d'octets, construit sur un service de livraison de datagrammes au mieux, fourni par le protocole Internet. TCP réalise cela en divisant les données en segments TCP, et en transportant ces segments dans des paquets IP. TCP garantit qu'une session TCP va retransmettre les segments TCP contenus dans tout paquet de données qui est perdu le long du chemin Internet entre les hôtes d'extrémité. TCP effectue normalement la retransmission en utilisant sa procédure de retransmission rapide, mais si la perte n'est pas détectée (ou si la retransmission ne réussit pas) TCP revient à une retransmission à temporisation de retransmission (RTO, *Retransmission Time Out*) en utilisant un temporisateur [RFC2581], [RFC2988]. (Les protocoles de liaison mettent aussi en œuvre des temporisateurs pour vérifier l'intégrité de la liaison, et pour aider l'ARQ de liaison.) TCP traite aussi toute duplication ou réarrangement de paquets introduit par le réseau IP. Il y a un certain nombre de variantes de TCP, avec des niveaux différents de sophistication dans leurs procédures du traitement de la récupération de pertes et de l'évitement d'encombrement. Loin d'être statique, le protocole TCP est lui-même l'objet de raffinements et d'évolutions graduels, par exemple [RFC2488], [RFC2760].

On peut raisonnablement s'attendre à ce que les réseaux Internet portent du trafic à partir d'une large gamme évolutive d'applications. Toutes les applications n'exigent pas d'utiliser ou de bénéficier du service fiable fourni par TCP. Dans l'Internet, ces applications sont portées par d'autres protocoles de transport, tels que le protocole de datagramme d'utilisateur (UDP, *User Datagram Protocol*) [RFC768].

### 1.1 ARQ de liaison

À la couche de liaison, ARQ fonctionne sur des blocs de données, qu'on appelle des trames, et tente de livrer les trames de la liaison d'envoi à la liaison receveuse sur un canal. Le canal fournit la connexion de couche physique sur laquelle fonctionne le protocole de liaison. Dans sa plus simple forme, un canal peut être une connexion directe de couche physique entre les deux nœuds de liaison (par exemple, à travers une longueur de câble ou sur un support sans fil). ARQ peut aussi être utilisé de bord à bord à travers un sous-réseau, lorsque le chemin comporte plus d'un support de couche physique. Les trames ont souvent une petite taille fixe ou maximum pour les besoins du traitement par les protocoles de commande d'accès au support (MAC, *Medium-Access Control*) et de liaison. Ceci contraste avec les longueurs variables de datagrammes IP, ou 'paquets'. Une trame de couche liaison peut contenir un paquet IP complet, ou une partie d'un paquet,

ou plusieurs paquet. Un mécanisme ARQ de liaison s'appuie sur une vérification d'intégrité pour chaque trame (par exemple, un CRC fort de couche de liaison [RFC3819]) pour détecter les erreurs de canal, et utilise un processus de retransmission pour retransmettre les trames perdues (c'est-à-dire, manquantes ou corrompues).

Les liaisons peuvent être bilatérales (*full-duplex*) (permettant une communication dans les deux directions sur des canaux séparés d'émission et de réception) unilatérales (*half-duplex*) (où une communication bidirectionnelle utilise un canal partagé d'émission et de réception, par exemple, IrDA, IEEE 802.11) ou unidirectionnelles (*simplex*) (où un seul canal ne permet la communication que dans une direction).

ARQ exige les deux chemins d'émission et de retour, et donc, la liaison ARQ peut être utilisée sur des liaisons qui emploient des liaisons bilatérales ou unilatérales. Lorsque un canal est partagé entre deux nœuds de liaison ou plus, un protocole MAC de liaison est nécessaire pour s'assurer que tous les nœuds qui exigent la retransmission peuvent obtenir l'accès au canal partagé. De tels schémas peuvent ajouter au délai (gigue) associé à la transmission de données en paquets et de trames de contrôle ARQ.

Lorsque on utilise ARQ sur une liaison où la probabilité de perte de trame est en rapport avec la taille de trame, il y a une taille de trame optimale pour tout taux d'erreur spécifique du canal cible. Pour permettre une utilisation efficace du canal, cette taille maximum de trame de liaison peut être considérablement inférieure à la taille maximum de datagramme IP spécifiée par l'unité de transmission maximum (MTU, *Maximum Transmission Unit*) IP. Chaque trame va alors contenir seulement une fraction de paquet IP, et on utilise une fragmentation transparente implicite du datagramme IP [RFC3819]. Une taille de trame plus petite introduit plus de redondance d'en-tête de trame par octet de charge utile transportée.

La fragmentation explicite à la couche réseau-IP est indésirable pour diverses raisons, et devrait être évitée [KEN87], [RFC3819]. Son utilisation peut être minimisée par l'utilisation de la découverte de la MTU du chemin [RFC1191], [RFC1435], [RFC1981].

Une autre façon de réduire le taux de perte de trame (ou de réduire la puissance du signal de transmission pour le même taux de perte de trame) est d'utiliser le codage, par exemple, la correction d'erreur directe (FEC, *Forward Error Correction*) [LIN93].

La FEC est couramment incluse dans la conception de la couche physique des liaisons sans fil et peut être utilisée simultanément avec l'ARQ de liaison. Des schémas de FEC qui combinent modulation et codage existent aussi, et peuvent également être adaptatifs. L'ARQ hybride [LIN93] combine la FEC adaptative avec les procédures d'ARQ de liaison pour réduire la probabilité de perte de trames retransmises. L'entrelaçage peut aussi être utilisé pour réduire la probabilité de perte de trame en dispersant plus largement les occurrences d'erreurs dans le canal afin d'améliorer la récupération d'erreurs ; cela rajoute des retards au délai de propagation existant du canal.

Le présent document n'examine pas l'utilisation de l'ARQ de liaison pour la prise en charge d'un service de diffusion ou de diffusion groupée au sein d'un sous-réseau, où une liaison peut envoyer un seul paquet à plus d'un receveur en utilisant une seule opération de transmission de liaison. Bien que de tels schémas soient pris en charge par certains sous réseaux, ils soulèvent un certain nombre de problèmes supplémentaires qui ne sont pas examinés ici.

Les liaisons qui prennent en charge la qualité de service (QS) fondée sur la réservation à états pleins conformément au modèle d'intégration de services (intserv) ne sont pas non plus explicitement discutées.

## 1.2 Causes des pertes de paquet sur une liaison

Tous les paquets envoyés sur une liaison ne sont pas nécessairement bien reçus par l'autre extrémité de la liaison. Il y a un certain nombre de causes qui rendent possible les pertes de paquet. Elles peuvent survenir lorsque les trames voyagent à travers une liaison, et incluent :

- Les pertes dues au bruit du canal, souvent caractérisées par une perte de trame aléatoire. Le bruit du canal peut aussi résulter d'autres conditions de dégradation du trafic sur le canal.
- Les pertes de trames dues aux interférences de canal. Ces interférences peuvent être aléatoires, structurelles, et dans certains cas, même périodiques.
- Une panne de liaison, période durant laquelle la liaison perd toutes ou virtuellement toutes les trames, jusqu'à ce que la liaison soit restaurée. C'est une caractéristique courante de certains types de liaisons, par exemple, du radio mobile cellulaire.

D'autres formes de perte de paquet sont sans rapport avec les conditions du canal, mais incluent :

- La perte d'une trame transmise dans un canal partagé où un protocole de MAC à capacité de contention est utilisé (par exemple, à cause de collisions). Ici, de nombreux protocoles exigent que la retransmission soit différée pour assurer la

- stabilité du canal partagé (c'est-à-dire, empêcher une excessive contention sur le canal (voir au paragraphe 1.5).
- ii. L'élimination de paquet à cause de l'encombrement. Les files d'attente vont finalement déborder lorsque le taux d'arrivée de nouveaux paquets à envoyer continue d'excéder le taux de transmission des paquets sortants sur la liaison.
  - iii. La perte due aux erreurs de mise en œuvre, y compris les fautes du matériel et les erreurs de logiciel. Ceci est reconnu comme une cause courante de corruption de paquet détectée chez les hôtes d'extrémité [STONE00].

Le taux de pertes et les schémas de pertes subis sont des fonctions de la conception des couches physique et de liaison. Elles varient de façon significative entre les différentes configurations de liaison. Les performances d'une mise en œuvre spécifique peuvent aussi varier considérablement sur la même configuration de liaison selon les différents types de canaux sur lesquels elles fonctionnent.

### 1.3 Pourquoi utiliser ARQ ?

Les raisons qui encouragent d'envisager l'utilisation d'ARQ incluent :

- a. ARQ à travers une liaison seule a une boucle de contrôle plus rapide que la boucle de contrôle des accusés de réception de TCP, qui a lieu sur le chemin plus long de bout en bout que celui sur lequel doit fonctionner TCP. Il est donc possible qu'ARQ fournisse une retransmission plus rapide des segments TCP perdus sur la liaison, au moins pour un nombre raisonnable de réessais [RFC3155], [SALT81].
- b. L'ARQ de liaison peut fonctionner sur des trames individuelles, en utilisant une fragmentation implicite transparente de liaison [RFC3819]. Les trames peuvent être beaucoup plus petites que les paquets IP, et la répétition de plus petites trames contenant des parties perdues ou erronées d'un paquet IP peut améliorer l'efficacité du processus d'ARQ et l'efficacité de la liaison.

Une procédure d'ARQ de liaison peut être capable d'utiliser des connaissances locales qui ne sont pas disponibles aux hôtes d'extrémité, pour optimiser les performances de livraison pour les conditions de liaison actuelles. Ces informations peuvent inclure des informations sur l'état de la liaison et du canal, par exemple, la connaissance du taux de transmission actuellement disponible, l'environnement d'erreurs prévalent, ou la puissance de transmission disponible dans les liaisons sans fils.

### 1.4 Techniques d'ARQ couramment utilisées

Un protocole d'ARQ de liaison utilise un mécanisme de protocole de liaison pour permettre à l'expéditeur de détecter les trames perdues ou corrompues et pour programmer leur retransmission. La détection des pertes de trames peut être via un temporisateur de protocole liaison, en détectant les trames d'accusé de réception positif manquantes, en recevant des trames explicites d'accusé de réception négatif, et/ou en interrogeant l'état du receveur de la liaison.

Quel que soit le mécanisme choisi, il y a deux catégories facilement décrites de processus de retransmission d'ARQ qui sont largement utilisées : l'ARQ d'arrêt et d'attente, et l'ARQ de fenêtre glissante.

#### 1.4.1 ARQ d'arrêt et attente

Un expéditeur qui utilise un ARQ à arrêt et attente (parfois appelé "ARQ paresseux" [LIN93]) transmet une seule trame et attend ensuite un accusé de réception du receveur pour cette trame. L'expéditeur soit continue la transmission avec la trame suivante, soit demande la répétition de la transmission de la même trame si l'accusé de réception indique que la trame d'origine a été perdue ou est corrompue.

L'ARQ à arrêt et attente est simple à mettre en œuvre, bien qu'inefficace, pour les concepteurs de protocoles, et est donc populaire, par exemple, TFTP [RFC1350] à la couche transport. Cependant, lorsque l'ARQ à arrêt et attente est utilisé dans la couche liaison, il ne convient bien que pour les liaisons qui ont un faible produit bande passante\*délat. Cette technique n'est pas examinée plus avant dans le présent document.

#### 1.4.2 ARQ de fenêtre glissante

Un protocole qui utilise l'ARQ de liaison à fenêtre glissante [LIN93] numérote chaque trame avec un numéro de séquence univoque, selon un processus modulaire. Le module définit la base de numérotation pour les numéros de séquence de trame, et la taille de l'espace de séquence. La plus grande valeur de numéro de séquence est vue par le protocole de liaison comme contiguë avec le premier (0), car l'espace de numérotation revient à zéro.

TCP est lui-même un protocole à fenêtre glissante à la couche transport [STE94], de sorte qu'on peut reconnaître des

similarités entre un protocole d'interface de liaison à interface de liaison et TCP de bout en bout. Un protocole de liaison à fenêtre glissante est beaucoup plus complexe à mettre en œuvre que le plus simple protocole à arrêt et attende décrit au paragraphe précédent, en particulier si l'ordre par flux est préservé.

À tout moment l'expéditeur se la liaison peut avoir un certain nombre de trames en cours qui attendent un accusé de réception, jusqu'à la limite de l'espace disponible dans sa fenêtre de transmission. Une fenêtre d'expéditeur de liaison suffisamment grande (équivalente ou supérieures au nombre de trames envoyés, ou supérieur à la capacité du produit bande passante\**délai de la liaison*) permet une continuation de la transmission de nouvelles trames. Une fenêtre d'expéditeur de liaison plus petite oblige l'expéditeur à faire une pause dans la transmission de nouvelles trames jusqu'à ce qu'une fin de temporisation ou une trame de contrôle, comme un accusé de réception, soit reçue. Lorsque des trames sont perdues, une plus grande fenêtre, c'est-à-dire, plus que le produit bande passante\**délai de la liaison*, est nécessaire pour permettre la continuation du fonctionnement pendant qu'à lieu la retransmission de trames.

Le module d'espace de numérotation détermine la taille du champ Numéro de séquence de l'en-tête de trame. Cet espace de numéros de séquence doit être plus grand que la taille de la fenêtre de liaison et, si on utilise l'ARQ à répétition sélective, plus grand que deux fois la taille de la fenêtre de liaison. Pour la continuation du fonctionnement, l'espace de numéros de séquence devrait être supérieur au produit de la capacité de la liaison et de la persistance de l'ARQ de liaison (exposée à la Section 2) afin que les trames en cours puissent être identifiées de façon univoque.

Comme avec TCP, qui fournit une fenêtre glissante de livraison sur la totalité du chemin de bout en bout plutôt que sur une seule liaison, il y a un grand nombre de variations à la mise en œuvre de base de fenêtre glissante, avec une complexité et une sophistication croissante pour l'adapter aux diverses conditions. La répétition sélective (SR, *Selective Repeat*) aussi appelée rejet sélectif (SREJ, *Selective Reject*), et le Go-Back-N, aussi appelé Rejet (REJ), sont des exemples de techniques d'ARQ qui utilisent des protocoles qui mettent en œuvre un ARQ à fenêtre glissante.

## 1.5 Causes de délai sur une liaison

Les liaisons et les protocoles de liaison contribuent au délai total de chemin subi entre les applications qui communiquent sur les hôtes d'extrémité. Le délai a de nombreuses causes, parmi lesquelles :

- la mise en file d'attente des paquets en entrée et la mise en mémoire tampon des trames à la tête de liaison avant la transmission sur le canal.
- le retard de retransmission, qui est un délai supplémentaire introduit pour les retransmissions par certains schémas de MAC lorsque ils fonctionnent sur un canal partagé pour prévenir une excessive rétention. Un fort niveau de rétention pourrait autrement survenir si, par exemple, un ensemble de receveurs de liaison sont tous retransmis immédiatement après une collision sur un canal partagé occupé. Les protocoles d'ARQ de liaison conçus pour les canaux partagés peuvent choisir un délai de retard, qui augmente avec le nombre de tentatives de retransmission d'une trame ; des analogies peuvent être remarquées avec l'évitement d'encombrement de bout en bout de TCP à la couche transport [RFC2581]. À l'opposé, une liaison sur un canal dédié (qui a une capacité pré-allouée à la liaison) peut envoyer une retransmission aussitôt que possible.
- l'attente de l'accès au canal alloué lorsque le canal est partagé. Il peut y avoir des retards induits par le traitement ou par le protocole avant qu'ait lieu la transmission [FER99], [PAR00].
- le processus de mise en série des trames et de leur transmission. Ce sont des fonctions de la taille de trame et de la vitesse de transmission de la liaison.
- le temps de propagation à la couche physique, limité par la vitesse de transmission du signal dans le support physique du canal.
- le traitement par trame, incluant le coût de la programmation de la QS, du chiffrement, de la FEC et de l'entrelacement. La FEC et l'entrelacement ajoutent aussi un délai substantiel, et dans certains cas, de la gigue supplémentaire. Les schémas d'ARQ de liaison hybrides, [LIN93] en particulier, peuvent subir des délais de traitement de receveur significatifs.
- le traitement de paquet, incluant de mettre en mémoire tampon le contenu des trames chez le receveur de la liaison pour le réassemblage du paquet, avant de procéder à la transmission du paquet.

Lorsque l'ARQ de liaison est utilisé, les étapes (b), (c), (d), (e), et (f) peuvent être répétées un certain nombre de fois, chaque fois que survient la retransmission d'une trame, augmentant le délai global pour le paquet transporté en partie par la trame. Les schémas d'ARQ adaptatif (par exemple, l'ARQ hybride utilisant les codes de FEC adaptative) peuvent aussi subir des traitements supplémentaires pour chaque trame retransmise.

Il est important de comprendre que les protocoles d'applications et de transport chez les hôtes d'extrémité ignorent tout des retards individuels apportés par chaque liaison du chemin, et ne voient que le délai global du chemin. Les performances de l'application sont donc déterminées par les retards accumulés dans la totalité du chemin Internet de bout en bout. Ce chemin peut comporter un nombre arbitraire ou même très fluctuant de liaisons, où toute liaison peut utiliser ou non l'ARQ.

Il en résulte qu'il n'est pas possible d'établir des limites fixes au délai acceptable qu'une liaison peut ajouter à un chemin ; d'autres liaisons du chemin vont ajouter un délai inconnu.

## 2. Persistance d'ARQ

Les protocoles d'ARQ peuvent être caractérisés par leur persistance. La persistance est la volonté du protocole de retransmettre les trames perdues pour assurer la livraison fiable du trafic à travers la liaison.

La persistance de retransmission d'une liaison définit pendant combien de temps il est permis à la liaison de retarder un paquet, pour tenter de transmettre toutes les trames qui portent le paquet et son contenu sur la liaison, avant d'abandonner et d'éliminer le paquet. Cette persistance peut normalement être mesurée en millisecondes, mais peut, si le délai de propagation de la liaison est spécifié, être exprimé en termes de nombre maximum de tentatives de retransmission permises à la liaison. Ce dernier ne se transpose pas toujours exactement en une limite de temps, pour les raisons expliquées au paragraphe 1.5.

Un exemple de protocole de liaison fiable qui est parfaitement persistant est le protocole HDLC de l'ISO dans le mode asynchrone équilibré (ABM, *Asynchronous Balanced Mode*) [ISO4335a].

Un protocole qui ne retransmet qu'un certain nombre de fois avant d'abandonner est moins persistant, par exemple, Ethernet [FER99], IEEE 802.11, ou GSM RLP [RFC2757]. Ici, une persistance inférieure assure aussi la stabilité et le partage équitable d'un canal partagé, même lorsque de nombreux envoyeurs tentent des retransmissions.

TCP, STCP [RFC2960] et un certain nombre d'applications qui utilisent UDP (par exemple, TFTP) mettent en œuvre leurs propres mécanismes de livraison fiable de bout en bout. De nombreuses applications TCP et UDP, par exemple, le multimédia en direct, bénéficient de livraison à temps de la part des couches inférieures avec une fiabilité suffisante, plutôt que d'une fiabilité parfaite avec des délais de liaison accrus.

### 2.1 Protocoles d'ARQ parfaitement persistants (fiables)

Un protocole d'ARQ parfaitement persistant est celui qui tente de fournir un service fiable, c'est-à-dire, une livraison dans l'ordre des paquets à l'autre extrémité de la liaison, sans paquet manquant ni dupliqué. Le protocole d'ARQ parfaitement persistant va répéter une trame perdue ou corrompue un nombre indéfini (et éventuellement infini) de fois jusqu'à ce que la trame soit bien reçue.

Si le trafic ne va pas plus loin à travers une autre liaison, et si il ne se produit pas de perte entre les hôtes d'extrémité, la persistance parfaite assure la fiabilité entre les deux extrémités de la liaison sans exiger de protocole de couche supérieure. Cette fiabilité peut devenir contreproductive pour le trafic qui traverse plusieurs liaisons, car il duplique et interagit avec les fonctionnalités des mécanismes de protocole des couches supérieures [SALT81].

Les arguments contre l'utilisation de la persistance parfaite du trafic IP incluent :

- a. le délai de liaison variable ; l'impact d'ARQ introduit un degré de gigue, fonction du délai de la couche physique et des temps de mise en série et de transmission de trames (discuté au paragraphe 1.5) sur tous les flux qui partagent une liaison qui effectue une retransmission de trame.
- b. la persistance parfaite ne donne pas une claire limite supérieure au délai maximum de retransmission pour la liaison. Des changements significatifs dans les délais de chemin causés par d'excessives retransmissions de liaison peuvent conduire à des fins de temporisation de temporisateurs de retransmission TCP, bien qu'une variance élevée du délai de liaison et du délai global de chemin résultant puisse aussi causer le choix d'une grande valeur de RTO TCP [LUD99b], [PAR00]. Cela va altérer le débit TCP, diminuant les performances globales, mais pour compenser, cela peut aussi diminuer la survenance de fins de temporisations dues à la poursuite de pertes de paquet.
- c. les applications qui ont besoin d'une livraison parfaitement fiable peuvent mettre en œuvre elles-mêmes une forme d'ARQ parfaitement persistante, ou utiliser un protocole de transport fiable chez les hôtes d'extrémité. La mise en œuvre de la persistance parfaite à chaque liaison le long du chemin entre les deux hôtes d'extrémité est redondante, mais ne peut pas assurer la même fiabilité que le transport de bout en bout [SALT81].
- d. l'ARQ de liaison ne devrait pas retarder le flux d'informations de contrôle de bout en bout. Par exemple, la retransmission ARQ de données pour un ou plusieurs flux ne devrait pas exagérément étendre les boucles de contrôle du protocole. Un délai excessif de duplicatas d'accusés de réception TCP (dupacks [STE94], [BAL97]) d'indicateurs de SACK, ou de notification explicite d'encombrement (ECN, *Explicit Congestion Notification*) va réduire la réactivité des flux TCP aux événements d'encombrement. Des questions similaires existent pour le contrôle de débit TCP convivial (TFRC, *TCP-Friendly Rate Control*) où le contrôle d'encombrement fondé sur les équations est utilisé avec UDP [RFC3448].

Les protocoles de liaison parfaitement persistants qui effectuent l'ARQ illimité, c'est-à-dire, qui continuent de retransmettre indéfiniment les trames jusqu'à ce qu'elles soient bien reçues, se trouvent rarement dans les mises en œuvre réelles.

La plupart des protocoles de liaison abandonnent en pratique la retransmission à un certain moment, mais ne le font pas nécessairement dans l'intention de limiter la persistance de retransmission ARQ. Un protocole peut, par exemple, terminer les retransmissions après un échec de connexion de la liaison, par exemple après qu'aucune trame n'a été bien reçue dans le délai d'un temporisateur pré-configuré. Le nombre de fois qu'un protocole retransmet une certaine trame (ou le nombre maximum de retransmissions) devient donc une fonction de nombreux paramètres différents (procédure ARQ, valeurs de temporisateur de la liaison, et procédure de surveillance de la liaison) plutôt que d'être pré-configuré.

Une autre caractéristique courante de ce type de comportement est que certaines mises en œuvre de protocole supposent que, après une défaillance de la liaison, les paquets mis en file d'attente pour être envoyés sur la liaison ne sont plus significatifs et peuvent être éliminés lors de l'abandon de la retransmission ARQ.

Des exemples de protocoles d'ARQ qui sont parfaitement persistants incluent le LAP-B de l'ISO/UIT-T [ISO7776] et le HDLC de l'ISO dans le mode asynchrone équilibré (ABM) [ISO4335a] qui utilise le rejet sélectif multiple (MSREJ, *Multiple Selective Reject*) [ISO4335b]. Ces protocoles vont retransmettre une trame un nombre illimité de fois jusqu'à réception de l'acquittement de la trame.

## 2.2 Protocoles d'ARQ à haute persistance (très fiables)

Les protocoles ARQ à haute persistance limitent le nombre de fois (ou le nombre de tentatives) que ARQ peut retransmettre une certaine trame avant que l'expéditeur abandonne la retransmission de la trame manquante et passe à la transmission des trames qui ont les numéros de séquence suivants dans la mémoire tampon. Cesser la retransmission d'une trame n'implique pas un manque de connectivité de la liaison et ne cause pas de changement d'état du protocole de liaison.

Il a été recommandé qu'un seul paquet IP ne devrait jamais être retardé par le réseau pendant plus que la durée de vie maximum de segment (MSL, *Maximum Segment Lifetime*) de 120 secondes définie pour TCP [RFC1122]. Il est cependant difficile en pratique de limiter le délai maximum d'un chemin Internet. Un cas où la durée de vie de segment (paquet) peut être significative est lorsque des solutions de remplacement de chemin de délais différents existent entre les hôtes d'extrémité et qu'on utilise une ingénierie de fluctuation de chemin ou de trafic insensible aux flux. Certains paquets TCP peuvent suivre un chemin court, tandis que d'autres suivent un chemin beaucoup plus long, par exemple, en utilisant un ARQ persistant sur une panne de liaison.

L'échec à limiter la durée de vie maximum de paquet peut résulter en un retour à zéro des numéros de séquence TCP à des taux de transmission élevés, et alors de vieux segments de données peuvent être confondus avec des segments plus récents si l'espace de numéros de séquence a été épuisé et réutilisé dans l'intervalle. Certaines mises en œuvre de TCP utilisent l'option de mesure de l'horodatage d'aller-retour (RTTM, *Round Trip Timestamp Measurement*) dans les paquets TCP pour lever cette ambiguïté, qui utilise l'algorithme de protection contre le retour à zéro des numéros de séquence (PAWS, *Protection Against Wrapped Sequence number*) [RFC1323].

En pratique, la MSL est habituellement très grande par rapport au RTO TCP normal. Le calcul du RTO TCP se fonde sur l'estimation du délai d'aller-retour du chemin [RFC2988]. Si le nombre de retransmissions sur la liaison cause un délai de chemin plus grand que la valeur du RTO, le temporisateur de retransmission TCP peut arriver à expiration, conduisant à une fin de temporisation et à la retransmission d'un segment (paquet) par l'expéditeur TCP.

Bien qu'une forte persistance soit bénéfique aux flux en vrac, le délai supplémentaire (et les variations de délai) qu'elle introduit peut être hautement indésirable pour les autres types de flux. Être capable de traiter les flux séparément, avec différentes classes de service liaison est utile, et c'est ce qui est exposé à la Section 3.

Les exemples de protocoles d'ARQ à persistance élevée incluent [BHA97], [ECK98], [LUD99a], [MEY99].

## 2.3 Protocoles d'ARQ à faible persistance (partiellement fiables)

Les caractéristiques d'une liaison qui utilise un protocole d'ARQ à faible persistance peuvent être résumés par :

- a. la liaison n'est pas parfaitement fiable et ne donne pas de garantie absolue de livraison, c'est-à-dire que l'émetteur va éliminer des trames lorsque il "abandonne" avant de recevoir un accusé de réception montrant la réussite de la transmission à travers la liaison.

- b. Il y a un abaissement de la limite de délai maximum ajouté que vont subir les paquets IP lorsque ils passent à travers la liaison (normalement inférieur au RTO de chemin TCP). Cela réduit l'interaction avec les temporisateurs TCP ou avec les schémas de contrôle d'erreur fondés sur UDP.
- c. La liaison offre une limite inférieure pour le temps pendant lequel la retransmission de toute trame peut bloquer l'achèvement de la transmission et de l'assemblage des autres paquets IP correctement et complètement reçus dont la transmission était commencée avant l'envoi de la trame manquante. Limiter le délai évite d'aggraver la rétention ou l'interaction entre différents flux de paquets (voir aussi au paragraphe 3.2).

Les exemples de protocoles d'ARQ à faible persistance incluent [SAM96], [WARD95], [CHE00].

## 2.4 Choisir sa persistance

Le RTO maximum de TCP est une limite supérieure du maximum de temps que va attendre TCP jusqu'à ce qu'il effectue une retransmission. La plupart des mises en œuvre de TCP vont généralement avoir un RTO TCP d'au moins plusieurs fois le délai de chemin.

Établi une persistance de liaison inférieure (par exemple, de l'ordre de 2 à 5 tentatives de retransmission) réduit les interactions potentielles avec le temporisateur de RTO TCP, et peut donc réduire la probabilité de copies dupliquées du même paquet présentes dans la mémoire tampon de transmission de la liaison dans certains schémas de perte.

Une liaison qui utilise une couche physique avec un faible délai de propagation va admettre des dizaines de tentatives de retransmission pour livrer une seule trame, et satisfaire quand même la limite de (b) du paragraphe 2.3. Dans ce cas, un faible délai est défini comme étant lorsque le temps total de transmission du paquet à travers la liaison est très inférieur à 100 ms (valeur courante de granularité du temporisateur interne du système TCP).

Un paquet peut traverser un certain nombre de liaisons successives sur son chemin total de bout en bout. C'est donc un argument pour une persistance bien inférieure sur toute liaison individuelle, car le délai dû à la persistance est accumulé le long du chemin que prend chaque paquet.

Certaines mises en œuvre ont choisi une persistance plus faible, revenant sur le jugement de TCP ou d'une application UDP pour retransmettre tout paquet qui n'est pas récupéré par le protocole ARQ de la liaison.

## 2.5 Impact sur les pannes de liaison

Les liaisons qui subissent des pertes persistantes, où de nombreuses trames consécutives sont corrompues pendant une longue période, peuvent aussi devoir être prises en considération. Des exemples de comportements du canal qui conduisent à des pannes de liaison incluent l'évanouissement, l'itinérance, et certaines formes d'interférences. En cas de perte, il y a une probabilité accrue qu'une demande de retransmission puisse être corrompue, et/ou une probabilité accrue qu'une trame retransmise soit aussi perdue. Ce type d'événement de perte est souvent appelé une "panne temporaire".

Si la panne temporaire dure plus longtemps que le RTO TCP, l'expéditeur TCP va aussi effectuer une retransmission de couche transport. En même temps, l'expéditeur TCP va réduire sa fenêtre d'encombrement (*cwnd*, *congestion window*) à un segment (paquet), recalculer son RTO, et attendre un paquet ACK. Si aucun accusé de réception n'est reçu, TCP va retransmettre encore, jusqu'à une limite des essais. TCP détermine seulement que la panne est terminée (c'est-à-dire, que la capacité du chemin est restaurée) en recevant un ACK. Si la persistance du protocole ARQ de liaison cause l'élimination de l'ACK par une liaison du chemin, l'expéditeur TCP doit attendre le prochain RTO de retransmission et son ACK pour apprendre que la liaison est restaurée. Cela peut être de nombreuses secondes après la fin de la panne temporaire.

Lorsque une couche de liaison est capable de différencier un ensemble de classes de service de liaison (voir au paragraphe 3.3) une persistance d'ARQ de liaison plus longue que le plus grand événement de perte de liaison peut être bénéfique pour une session TCP. Cela permettrait à TCP de restaurer rapidement la transmission sans avoir besoin d'attendre pendant une temporisation de retransmission, améliorant généralement les performances de TCP face à des pannes temporaires. Les mises en œuvre de tels schémas font l'objet de travaux d'étude.

Lorsque survient une panne pour un expéditeur qui partage un canal commun avec d'autres nœuds, une persistance non contrôlée peut continuer à consommer des ressources de transmission pendant la durée de la panne. Cela peut être indésirable, car cela réduit la capacité disponible pour les autres nœuds qui partagent le canal, et qui ne subissent pas nécessairement la même panne. Ces nœuds pourraient autrement utiliser le canal pour des transferts plus productifs. Dans de tels cas, la persistance est souvent limitée par d'autres mécanismes de contrôle. Pour contrer de tels effets de contention, les protocoles d'ARQ peuvent retarder les demandes de retransmission, ou différer la retransmission des trames demandées

jusqu'à ce que la panne se termine pour l'expéditeur.

Une approche suggérée comme solution de remplacement pour une couche liaison qui est capable d'identifier des flux séparés est d'utiliser la persistance de liaison (paragraphe 2.3) avec un mécanisme de couche supérieure, par exemple un mécanisme qui tente de livrer un paquet (ou un segment TCP entier) par flux TCP après un événement de perte [RFC3819]. Cela est destiné à assurer que la transmission TCP est restaurée rapidement. Les algorithmes pour mettre en œuvre cela font encore l'objet d'études.

### 3. Traitement des paquets et des flux

#### 3.1 Ordre des paquets

Un débat courant porte sur la question de savoir si il devrait être permis à une liaison de transmettre des paquets dans un ordre différent de celui dans lequel ils ont été originellement reçus à son interface de transmission.

Les réseaux IP ne sont pas obligés de livrer tous les paquets IP dans l'ordre, bien que dans la plupart des cas, les réseaux livrent bien les paquets dans leur ordre de transmission d'origine. Les routeurs qui prennent en charge la mise en file d'attente fondée sur la classe réarrangent effectivement l'ordre des paquets reçus, en réarrangeant les paquets dans des flux différents, mais cela conserve habituellement l'ordre par flux.

La mise en file d'attente fondée sur la politique, qui permet un accès plus équitable à la liaison, peut aussi réarranger les paquets. Il y a encore de grands débats sur les meilleurs algorithmes, et sur la taille optimale de file d'attente pour les différentes vitesses de liaison. Cela est cependant sans rapport avec l'utilisation de l'ARQ de liaison et s'applique à tous routeurs qui est un (potentiel) goulet d'étranglement.

Bien que de petites quantités de réarrangements soient courantes dans les réseaux IP [BEN00], un réarrangement significatif n'est pas souhaitable au sein d'un flux car il peut avoir un certain nombre d'effets :

- a. un réarrangement va augmenter la gigue de paquet pour les applications en temps réel. Cela peut conduire à des pertes de données d'application si une mémoire tampon d'exécution petite est utilisée par l'application receveuse ;
- b. un réarrangement va entrelacer les arrivées de segments TCP, conduisant à la génération d'ACK dupliqués (dupacks), conduisant à des hypothèses de perte. La réception d'un ACK suivie par une séquence de trois dupacks identiques cause chez l'expéditeur TCP le déclenchement de la retransmission et récupération rapide, une forme d'évitement d'encombrement, car TCP suppose toujours que la perte de paquet est due à l'encombrement [RFC2581], [STE94]. Cela réduit l'efficacité du débit de TCP pour autant que l'application soit concernée, bien que cela ne doive pas impacter l'intégrité des données.

De plus, un réarrangement peut avoir un impact négatif sur le traitement par certaines mises en œuvre maladroites des piles TCP/IP existantes, en conduisant à des effets collatéraux non désirés dans les mises en œuvre maladroites de codes de réassemblage de fragment IP, de codes de démultiplexage (filtre) IP, ou d'applications UDP.

Les effets d'un réarrangement doivent aussi être pris en compte lorsque ils cassent le paradigme de bout en bout et dans l'évaluation des relais de couche transport tels que les mises en œuvre de TCP partagées ou de mandataires d'amélioration de protocole [RFC3135].

Comme avec le délai de chemin total, les flux TCP et UDP sont impactés par l'effet cumulatif des réarrangements le long du chemin entier. Les concepteurs de protocole de liaison ne doivent pas supposer que leur liaison est la seule à entreprendre des réarrangements de paquets, car un certain niveau de réarrangement peut être introduit par d'autres liaisons le long du même chemin, ou par des traitements de routeurs au sein du réseau [BEN00]. Idéalement, le protocole de liaison ne devrait pas contribuer à des réarrangements au sein d'un flux, ou au moins on devrait s'assurer qu'il n'augmente pas de façon significative le niveau de réarrangement au sein du flux. Pour réaliser cela, une mise en mémoire tampon est nécessaire chez la liaison receveuse. La quantité totale de mémoire tampon requise est fonction du produit bande passante\*délai de la liaison et du niveau de persistance d'ARQ, et est limitée par la taille de fenêtre de la liaison.

Un certain nombre de protocoles expérimentaux d'ARQ ont permis des livraisons en désordre [BAL95], [SAM96], [WARD95].

### 3.2 Utilisation de l'ARQ de liaison pour la prise en charge de flux multiples

On peut s'attendre à ce que la plupart des liaisons portent plus d'un flux IP à la fois. Certaines liaisons à forte capacité sont supposées porter un très grand nombre de flux simultanés, souvent de et vers un grand nombre d'hôtes d'extrémité différents. Avec l'utilisation de plusieurs applications chez un hôte d'extrémité, plusieurs flux peuvent être considérés comme la norme plutôt que l'exception, même pour les liaisons de dernier bond.

Lorsque des paquets provenant de plusieurs flux sont simultanément en transit au sein d'un protocole ARQ de liaison, ARQ peut causer un certain nombre d'effets supplémentaires :

- a. ARQ introduit un délai variable (gigue) chez un flux TCP qui partage une liaison avec un autre flux qui subit des pertes. Ce délai supplémentaire, introduit par le besoin qu'à une liaison de fournir une livraison en séquence des paquets, peut avoir un impact négatif sur les autres applications qui partagent la liaison, et peut augmenter la durée de la période initiale de démarrage lent pour les flux TCP de ces applications. Ceci est significatif pour les flux TCP à courte durée (par exemple, ceux utilisés par HTTP/1.0 et avant) qui passent la plus grande partie de leur vie en démarrage lent.
- b. ARQ introduit de la gigue chez les flux UDP qui partagent une liaison avec un autre flux sujet à des pertes. Un protocole de bout en bout peut ne pas exiger de livraison fiable pour ses flux, en particulier si il prend en charge une application sensible au délai.
- c. L'ARQ à forte persistance peut retarder les paquets assez longtemps pour causer la fin de temporisation prématurée du temporisateur de RTO d'un autre flux TCP, bien que les mises en œuvre modernes de TCP ne devraient pas subir cela car leurs valeurs de RTO calculées devraient laisser une marge suffisante sur les RTT de chemin pour s'accommoder d'une quantité raisonnable de gigue.

Le réarrangement des paquets qui appartiennent à des flux différents peut être souhaitable [LUD99b], [CHE00] pour réaliser un partage équitable de la liaison entre les sessions établies de transfert de données en vrac et les sessions qui transmettent moins de données, mais bénéficieraient de moindres délais de transit sur la liaison. Préserver l'ordre au sein de chaque flux individuel, pour éviter les effets du réarrangement décrit plus haut au paragraphe 3.1, en vaut la peine.

### 3.3 Différenciation des classes de service de liaison

On considère généralement qu'une forte persistance de ARQ ne convient pas pour de nombreuses applications qui utilisent UDP, lorsque la livraison fiable n'est pas toujours requise et lorsque elle peut introduire une gigue inacceptable, mais qu'elle peut être bénéfique pour les transferts de données en vrac sous certaines conditions de liaison. Un schéma qui différencie les flux de paquets en deux classes ou plus, pour fournir un service différent à chaque classe, est donc souhaitable.

L'observation du comportement des flux peut nous dire quels flux sont contrôlés et sensibles à l'encombrement, ou incontrôlés et non sensibles, de sorte qu'on puisse les traiter différemment et assurer l'équité. Cependant, cela ne peut pas nous dire si un flux est destiné à être fiable ou non par son application, ou ce que l'application requiert pour le meilleur fonctionnement.

Prendre en charge des services de liaison différents pour différentes classes de flux exige donc que la liaison soit capable de distinguer les différents flux l'un de l'autre. Cela nécessite généralement une indication explicite de la classe associée à chaque flux.

Certains schémas potentiels pour indiquer la classe d'un paquet incluent :

- a. d'utiliser les bits de type de service (ToS) dans l'en-tête IP [RFC791]. L'IETF a remplacé ces bits globalement définis, qui n'étaient pas largement utilisés, par le modèle de services différenciés (diffserv [RFC2475], [RFC3260]). Dans diffserv, chaque paquet porte un codet de service différencié (DSCP, *Differentiated Service Code Point*) qui indique à quelle classe de l'ensemble diffserv appartient le flux. Chaque routeur transpose la valeur de DSCP d'un paquet IP reçu en une de l'ensemble des comportements par bond (PHB, *Per Hop Behaviour*) lorsque le paquet est traité au sein du réseau. Les utilisations de Diffserv incluent l'acheminement fondé sur la politique, la mise en file d'attente fondée sur la classe, et la prise en charge d'autres métriques de QS, incluant la priorité de paquet IP, le délai, la fiabilité, et le coût.
- b. d'inspecter l'en-tête de paquet réseau et de voir le type de protocole IP [RFC791] pour obtenir une idée du protocole de transport utilisé et donc de deviner ses besoins. Ceci n'est pas possible lorsque on porte une charge utile chiffrée, par exemple, en utilisant les extensions de sécurité IP (IPsec) avec le chiffrement de charge utile de l'encapsulation de charge utile de sécurité (ESP, *Encapsulation Security Payload*) [RFC2406].
- c. d'inspecter les informations de l'en-tête de paquet de transport pour voir les en-têtes TCP ou UDP et les numéros d'accès (par exemple, [PAR00], [BAL95]). Ceci n'est pas possible lorsque on utilise le chiffrement de charge utile, par exemple, IPsec avec le chiffrement de charge utile ESP [RFC2406], et qu'on subit la redondance de traitement pour chaque paquet envoyé sur la liaison.

Il y a cependant quelques inconvénients à ces schémas :

- i. Les valeurs de codets de ToS/services différenciés (DSCP) [RFC2475] peuvent n'être pas établies de façon fiable, et peuvent être outrepassées par des routeurs intermédiaires le long du chemin du paquet. Ces valeurs peuvent être établies par un FAI, et ne pas nécessairement indiquer le niveau de fiabilité requis par l'application d'extrémité. La liaison doit être configurée en connaissant la signification locale des valeurs.
- ii. Le tunnelage du trafic (par exemple, GRE, MPLS, L2TP, encapsulation IP dans IP) peut agréger les flux de différentes classes de transport, compliquant la classification des flux individuels avec les schémas (b) et (c) et supportant des traitements d'en-tête supplémentaires si le contenu du tunnel est inspecté.
- iii. L'utilisation du numéro d'accès TCP/UDP fait des hypothèses sur le comportement et les exigences de l'application. De nouvelles applications ou de nouveaux protocoles peuvent invalider ces hypothèses, comme peut le faire par exemple, la traduction d'accès-adresse réseau, où les numéros d'accès sont retransposés [RFC3022].
- iv. Dans IPv6, l'en-tête IPv6 entier doit être analysé pour localiser le protocole de couche transport, ce qui ajoute de la complexité à l'inspection d'en-tête. Là encore, cela suppose qu'on n'utilise pas le chiffrement de charge utile IPsec.

En dépit des difficultés pour fournir un cadre pour une identification précise des flux, l'approche (a) peut être avantageuse, et est préférable à l'ajout d'optimisations qui sont déclenchées par l'inspection du contenu de paquets IP spécifiques. Certaines de ces optimisations sont discutées en détail dans [LUD99b].

La gestion des flux est désirable ; une claire identification des flux augmente le nombre d'outils disponibles pour le concepteur de liaisons, et permet des stratégies d'ARQ plus complexes qui pourraient autrement faire de mauvaises hypothèses sur les exigences et les comportements du trafic lorsque l'identification des flux n'est pas faite.

Les liaisons ne sont pas capables de distinguer clairement et en toute sécurité entre les flux sensibles au délai, par exemple, le multimédia en temps réel, les interrogations au DNS ou telnet, et les flux non sensibles au délai, comme par exemple, le transfert ftp en vrac ou les transferts fiables en diffusion groupée, et elles ne peuvent pas séparer en toute sûreté les classes de service liaison. Tous les flux devraient donc subir le même comportement de liaison.

En général, si la séparation des flux selon la classe n'est pas praticable, une faible persistance est le mieux pour l'ARQ de liaison.

## 4. Conclusions

Un certain nombre de techniques peuvent être utilisées par les concepteurs de protocole de liaison pour contrer les effets des erreurs ou des pertes du canal. Une de ces techniques est la demande de répétition automatique (ARQ, *Automatic Repeat ReQuest*) qui a été et continue d'être utilisée sur les liaisons qui portent du trafic IP. Un protocole ARQ retransmet les trames de liaison qui ont été corrompues durant la transmission à travers un canal. L'ARQ de liaison peut améliorer significativement la probabilité de réussite de la transmission des paquets IP sur des liaisons enclines à des pertes de trame occasionnelles.

Un plus faible taux de perte de paquet bénéficie généralement aux protocoles de transport et aux applications d'hôte d'extrémité. Les applications qui utilisent TCP tirent généralement avantage des chemins Internet qui ont peu ou pas de pertes et de faibles délais d'aller-retour de chemin. Cela réduit l'impact sur les applications, permet une croissance plus rapide de la fenêtre d'encombrement de TCP durant la phase de démarrage lent, et assure une prompt réaction aux échanges de protocole de bout en bout (par exemple, aux retransmissions, aux indications d'encombrement). Les applications qui utilisent d'autres protocoles de transport, par exemple, UDP ou SCTP, tirent aussi avantage de faibles pertes et d'une livraison à temps.

Un effet collatéral de l'ARQ est que le délai de transit de liaison est augmenté quand des trames sont retransmises. À de faibles taux d'erreur, de nombreux détails d'ARQ, comme le degré de persistance ou d'éventuelles livraisons déclassées deviennent sans importance. La plupart des pertes de trames seront résolues en une ou deux tentatives de retransmission, et il est généralement peu probable que cela ait un impact significatif sur, par exemple, TCP. Cependant, sur des liaisons partagées à fort délai, l'impact d'ARQ sur les autres flux UDP ou TCP peut conduire à une gigue indésirable.

Lorsque le taux d'erreurs est très variable, une forte persistance d'ARQ de liaison peut donner de bonnes performances pour un seul flux TCP. Cependant, des interactions entre flux peuvent survenir lorsque de nombreux flux partagent les capacités de la même liaison. Une procédure d'ARQ de liaison qui assure la gestion des flux va être profitable. Une moindre persistance d'ARQ a aussi des mérites, et est préférable pour les applications qui utilisent UDP. Le raisonnement est ici que la liaison peut effectuer un travail utile de transmission de quelques paquets complets, et que bloquer tous les flux en retransmettant les trames d'un seul paquet avec une forte persistance n'est pas souhaitable.

Durant une panne de liaison, des interactions entre ARQ et plusieurs flux sont moins significatives ; le protocole ARQ ne

va vraisemblablement également pas réussir à retransmettre des trames pour tous les flux. Une forte persistance peut avantager les flux TCP en permettant une prompte récupération une fois le canal restauré.

Une faible persistance d'ARQ est particulièrement utile lorsque il est difficile ou impossible de classer les flux de trafic, et donc chaque flux de façon indépendante, et lorsque la capacité de la liaison peut s'accommoder d'un grand nombre de flux simultanés.

Les concepteur d'ARQ de liaison devraient envisager les implications de leur concept sur la totalité de l'Internet. Des effets comme un délai de transit accru, la gigue, et le réarrangement sont cumulatifs lorsque effectués sur plusieurs liaisons le long d'un chemin Internet. Il est donc très difficile de dire combien de liaisons ARQ peuvent exister en série le long d'un chemin Internet arbitraire entre des hôtes d'extrémité, en particulier lorsque le chemin suivi et ses liaisons peuvent changer d'un moment à l'autre.

En résumé, lorsque les liaisons ne peuvent pas classer les flux de trafic et les traitent séparément, une faible persistance est généralement souhaitable ; préserver l'ordre des paquets est généralement souhaitable. Une persistance extrêmement forte et une persistance parfaite ne sont généralement pas désirables ; l'ARQ à forte persistance est une mauvaise idée sauf si la classification des flux et une connaissance détaillée et précise des exigences des flux rendent possible de déployer une forte persistance lorsque elle sera avantageuse.

On a actuellement pas assez d'expérience pour recommander un schéma spécifique d'ARQ pour une classe de liaisons. Il est aussi important de réaliser que l'ARQ de liaison est simplement une méthode de récupération d'erreur, et que d'autres techniques complémentaires de couche physique peuvent être utilisées à la place d'ARQ, ou avec elle, pour améliorer le débit global de la liaison pour le trafic IP.

Le choix de schémas potentiels inclut d'adapter le débit de données, d'adapter la bande passante de signal, d'adapter la puissance d'émission, la modulation adaptative, l'adaptation de la redondance des informations / du contrôle d'erreurs directes, et l'entrelacement. Tous ces schémas peuvent être utilisés pour améliorer l'énergie du signal reçu par bit, et donc réduire les erreurs, la perte de trame et les taux de perte de paquet résultants étant données les conditions spécifiques du canal.

Des recherches supplémentaires doivent être effectuées pour identifier plus clairement l'importance d'un compromis entre les questions posées ci-dessus sur les divers types de liaison et sur les divers types de canaux. Il serait utile que les chercheurs et les développeurs indiquent clairement le modèle de perte, la capacité et les caractéristiques de liaison, les délais de liaison et de chemin de bout en bout, les détails de TCP, et le nombre (et les détails) des flux qui partagent une liaison lorsque ils décrivent leurs expériences. Dans chaque cas, il est recommandé que les détails spécifiques des caractéristiques et mécanismes de la liaison soient pris en compte ; les solutions varient avec les conditions.

## **5. Considérations pour la sécurité**

Aucune implication pour la sécurité n'a été identifiée comme impactant directement le trafic IP. Cependant, un service de liaison non fiable peut avoir un impact négatif sur certains protocoles existants de distribution et de gestion de clés de couche liaison des données si le chiffrement de la liaison est aussi utilisé sur la liaison.

Des attaques de déni de service exploitant le comportement du protocole de liaison, par exemple, en utilisant la connaissance de son comportement de retransmission et le délai de propagation pour causer une forme particulière d'embrouillage, peuvent être spécifiques d'un scénario de liaison individuel.

## **6. Considérations relatives à l'IANA**

Aucune allocation n'est demandée à l'IANA.

## **7. Remerciements**

Une grande partie de ce qui est décrit ici a été développé à partir d'un résumé d'un sous ensemble des discussions archivées de la liste de diffusion PILC de l'IETF. Nous remercions les contributeurs de PILC de leurs vigoureux débats.

En particulier, les auteurs tiennent à remercier Spencer Dawkins, Aaron Falk, Dan Grossman, Merkourios Karaliopoulos, Gary Kenward, Reiner Ludwig et Jean Tourrilhes de leurs commentaires détaillés.

## 8. Références

Les références de la forme RFCnnnn sont des documents de demandes de commentaires pour l'Internet (RFC, *Request for Comments*) disponibles en ligne à <http://www.rfc-editor.org/>.

### 8.1 Références normatives

- [RFC0768] J. Postel, "Protocole de [datagramme d'utilisateur](#) (UDP)", (STD 6), 28 août 1980.
- [RFC0791] J. Postel, éd., "Protocole Internet - Spécification du [protocole du programme Internet](#)", STD 5, septembre 1981.
- [RFC0793] J. Postel (éd.), "Protocole de [commande de transmission](#) – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.
- [RFC1122] R. Braden, "[Exigences pour les hôtes Internet](#) – couches de communication", STD 3, octobre 1989. (*MàJ par la RFC6633*)
- [RFC2406] S. Kent et R. Atkinson, "Encapsulation de [charge utile de sécurité](#) IP (ESP)", novembre 1998. (*Obsolète, voir RFC4303*)
- [RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang et W. Weiss, "[Architecture pour services différenciés](#)", décembre 1998. (*MàJ par RFC3260*)
- [RFC2581] M. Alman, V. Paxson et W. Stevens, "[Contrôle d'encombrement avec TCP](#)", avril 1999. (*Obsolète, voir RFC5681*)
- [RFC2988] V. Paxson, M. Allman, "Calcul du temporisateur de retransmission de TCP", novembre 2000. (*P.S.)(Obsolète, voir RFC6298*)
- [RFC3135] J. Border et autres, "Amélioration des performances des mandataires destinée à atténuer les dégradations concernant la liaison", juin 2001. (*Information*)
- [RFC3260] D. Grossman, "Nouvelle [terminologie et précisions pour Diffserv](#)", avril 2002. (*Information*)

### 8.2 Références pour information

- [BAL95] Balakrishnan, H., Seshan, S. and R. H. Katz, "Improving Reliable Transport and Handoff Performance n Cellular Wireless Networks", ACM MOBICOM, Berkeley, 1995.
- [BAL97] Balakrishnan, H., Padmanabhan, V. N., Seshan, S. and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ACM Transactions on Networking, 5(6), pp. 756-759, 1997.
- [BEN00] Bennett, J. C., Partridge, C. and N. Schectman, "Packet Reordering is Not Pathological Network Behaviour", IEEE/ACM Transactions on Networking, 7(6), pp. 789-798, 2000.
- [BHA97] Bhagwat, P., Bhattacharya, P., Krishna A. and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs", ACM/Baltzer Wireless Networks Journal, (3)1, 1997.
- [CHE00] Cheng, H. S., G. Fairhurst et al., "An Efficient Partial Retransmission ARQ Strategy with Error Codes by Feedback Channel", IEE Proceedings - Communications, (147)5, pp. 263-268, 2000.
- [ECK98] Eckhardt, D. A. and P. Steenkiste, "Improving Wireless LAN Performance via Adaptive Local Error Control", IEEE ICNP, 1998.
- [FER99] Ferrero, A., "The Eternal Ethernet", Addison-Wesley, 1999.

- [ISO4335a] International Standardization Organization, "HDLC Procedures: Specification for Consolidation of Elements of Procedures", norme ISO 4335 et Amendement 1, 1985.
- [ISO4335b] International Standards Organization, "HDLC Procedures: Elements of Procedures, Amendment 4: Multi-Selective Reject Option", ISO 4335/4, 1991.
- [ISO7776] International Standards Organization, "Specification for X.25 LAPB-Compatible DTE Data Link Procedures", ISO 7776, 1985.
- [KEN87] Kent, C. A. and J. C. Mogul, "Fragmentation Considered Harmful", Proceedings of ACM SIGCOMM 1987, ACM Computer Communications Review, 17(5), pp. 390-401, 1987.
- [LIN93] Lin, S. and D. Costello, "Error Control Coding: Fundamentals and Applications", Prentice Hall, 1993.
- [LUD99a] Ludwig, R., Rathonyi, B., Konrad, A., Oden, K., and A. Joseph, "Multi-Layer Tracing of TCP over a Reliable Wireless Link", ACM SIGMETRICS, pp. 144-154, 1999.
- [LUD99b] Ludwig, R., Konrad, A., Joseph, A. and R. H. Katz, "Optimizing the End-to-End Performance of Reliable Flows over Wireless Links", ACM MobiCOM, 1999.
- [MEY99] Meyer, M., "TCP Performance over GPRS", IEEE Wireless Communications and Networking Conference, 1999.
- [PAR00] Parsa, C. and J. J. Garcia-Luna-Aceves, "Improving TCP Performance over Wireless Networks at the Link Layer", ACM Mobile Networks and Applications Journal, (5)1, pp. 57-71, 2000.
- [RFC1191] J. Mogul et S. Deering, "[Découverte de la MTU](#) de chemin", novembre 1990.
- [RFC1323] V. Jacobson, R. Braden et D. Borman, "[Extensions TCP](#) pour de bonnes performances", mai 1992.
- [RFC1350] K. Sollins, "[Protocole TFTP](#) (révision 2)", STD 33, juin 1992. (*MàJ par 1782-85, 2347\_49*)
- [RFC1435] S. Knowles, "Avis de l'IESG d'une expérience avec découverte de la MTU de chemin", mars 1993. (*Info*)
- [RFC1981] J. McCann, S. Deering, J. Mogul, "Découverte de la [MTU de chemin pour IP version 6](#)", août 1996. (*D.S.*)
- [RFC2488] M. Allman et autres, "Amélioration de TCP sur canal satellite avec les mécanismes standard", janvier 1999. ([BCP0028](#))
- [RFC2757] G. Montenegro et autres, "Longs réseaux fins", janvier 2000. (*Information*)
- [RFC2760] M. Allman et autres, "Recherches en cours sur TCP concernant les satellites", février 2000. (*Information*)
- [RFC2960] R. Stewart et autres, "Protocole de transmission de commandes de flux", octobre 2000. (*Obsolète, voir [RFC4960](#) (P.S.)*)
- [[RFC3022](#)] P. Srisuresh, K. Egevang, "[Traducteur d'adresse réseau IP traditionnel](#)", janvier 2001. (*Information*)
- [RFC3155] S. Dawkins et autres, "Implications des [liaisons avec des erreurs sur les performances](#) de bout en bout", août 2001. ([BCP0050](#))
- [RFC3448] M. Handley, S. Floyd, J. Padhye, J. Widmer, "Contrôle de débit convivial sur TCP (TFRC) : Spécification du protocole", janvier 2003. (*Obsolète, voir [RFC5348](#) (P.S.)*)
- [RFC3819] P. Karn et autres, "Conseils aux concepteurs de sous-réseaux Internet", juillet 2004. ([BCP0089](#))
- [SALT81] Saltzer, J. H., Reed, D. P. and D. Clark, "End-to-End Arguments in System Design", Second International Conference on Distributed Computing Systems, pp. 509-512, 1981. Publié avec des modifications mineures dans ACM Transactions in Computer Systems (2)4, pp. 277-288, 1984.
- [SAM96] Samaraweera, N. and G. Fairhurst, "Robust Data Link Protocols for Connection-less Service over Satellite

Links", International Journal of Satellite Communications, 14(5), pp. 427-437, 1996.

- [SAM98] Samaraweera, N. and G. Fairhurst, "Reinforcement of TCP/IP Error Recovery for Wireless Communications", ACM Computer Communications Review, 28(2), pp. 30-38, 1998.
- [STE94] Stevens, W. R., "TCP/IP Illustrated, Volume 1", Addison-Wesley, 1994.
- [STONE00] Stone, J. and C. Partridge, "When the CRC and TCP Checksum Disagree", Proceedings of SIGCOMM 2000, ACM Computer Communications Review 30(4), pp. 309-321, septembre 2000.
- [WARD95] Ward, C., and al., "A Data Link Control Protocol for LEO Satellite Networks Providing a Reliable Datagram Service", IEEE/ACM Transactions on Networking, 3(1), 1995.

## Adresse des auteurs

Godred Fairhurst  
Department of Engineering  
University of Aberdeen  
Aberdeen AB24 3UE  
United Kingdom  
mél : [gorry@erg.abdn.ac.uk](mailto:gorry@erg.abdn.ac.uk)  
<http://www.erg.abdn.ac.uk/users/gorry/>

Lloyd Wood  
Cisco Systems Ltd  
4 The Square  
Stockley Park  
Uxbridge UB11 1BY  
United Kingdom  
EMail: [lwood@cisco.com](mailto:lwood@cisco.com)  
<http://www.ee.surrey.ac.uk/Personal/L.Wood/>

## Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2002). Tous droits réservés.

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

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

Le présent document et les informations 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.