Groupe de travail Réseau
Request for Comments : 4340
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

E. Kohler, UCLA
M. Handley, UCL
S. Floyd, ICIR
mars 2006

Protocole de contrôle d'encombrement de datagrammes (DCCP)

Statut de ce mémoire

Ce document spécifie un protocole de normalisation Internet pour la discussion de la communauté Internet, et demande des suggestions pour des améliorations. Prière de se référer à l'édition actuelle de "Normes officielles des protocoles de l'Internet "(STD 1) pour l'état de la normalisation et le statut de ce protocole. La distribution de ce mémoire est illimitée.

Résumé

Le protocole de contrôle d'encombrement de datagrammes (DCCP, *Datagram Congestion Control Protocol*) est un protocole de transport qui fournit des connexions bidirectionnelles en envoi individuel de datagrammes non fiables à encombrement contrôlé. DCCP est adapté pour des applications qui transfèrent des quantités très importantes de données et qui peuvent bénéficier d'un contrôle sur le compromis entre délais et fiabilité.

Table des matières

Liste des tableaux

1. Présentation

Le protocole de contrôle d'encombrement de datagrammes (DCCP, *Datagram Congestion Control Protocol*) est un protocole de transport qui met en œuvre des connexions bidirectionnelles en envoi individuel de datagrammes non fiables à encombrement contrôlé. Plus précisément, DCCP fournit les éléments suivants :

- o flux non fiables de datagrammes.
- o prises de contact fiables pour l'installation et la suppression de connexion.
- o négociation fiable d'options, y compris la négociation d'un mécanisme convenable de contrôle d'encombrement.
- o mécanismes permettant aux serveurs d'éviter la conservation de l'état pour les tentatives de connexion non acquittées et les connexions déjà terminées.
- o contrôle d'encombrement incorporant la notification explicite d'encombrement (ECN, *Explicit Congestion Notification*) [RFC3168] et le nom occasionnel ECN [RFC3540].
- o des mécanismes d'accusé de réception qui communiquent les pertes de paquet et les informations d'ECN. Les accusés de réception sont transmis de manière aussi fiable que l'exigent les mécanismes pertinents de contrôle d'encombrement, éventuellement de manière complètement fiable.
- o des mécanismes facultatifs qui disent à l'application d'envoi, avec une grande fiabilité, quels paquets de données ont atteint le receveur, et si ces paquets ont été marqués ECN, corrompus, ou abandonnés dans la mémoire tampon de réception.
- o la découverte de l'unité maximum de transmission de chemin (PMTU, *Path Maximum Transmission Unit*) [RFC1191].
- o un choix de mécanismes modulaires de contrôle d'encombrement. Deux mécanismes sont actuellement spécifiés : le contrôle d'encombrement de style TCP [RFC4341] et le contrôle en douceur de débit TCP (TFRC, *TCP-Friendly Rate Control*) [RFC4342]. DCCP est facilement extensible à d'autres formes de contrôle d'encombrement en envoi individuel.

DCCP est destiné à des applications telles que le support de direct qui peut bénéficier d'un contrôle sur les arbitrages entre le retard et la fiabilité dans l'ordre de livraison. TCP n'est pas bien adapté pour ces applications, car la fiabilité dans l'ordre de livraison et le contrôle d'encombrement peuvent provoquer des retards arbitrairement longs. UDP évite de longs retards, mais les applications UDP qui mettent en œuvre le contrôle d'encombrement doivent le faire d'elles mêmes. DCCP fournit un contrôle d'encombrement incorporé, notamment la prise en charge de ECN, pour les flux de datagrammes non fiables, en évitant les retards arbitraires associés au protocole TCP. Il met également en œuvre l'établissement de connexions fiables, et la négociation de caractéristiques.

2. Justification du concept

Un objectif de conception de DCCP était de donner à la plupart des applications UDP en flux continus peu de raisons de ne pas passer à DCCP, une fois qu'il est déployé. Pour faciliter cette démarche, DCCP a été conçu pour avoir aussi peu de frais généraux que possible, tant en termes de taille d'en-tête de paquet qu'en termes de redondances d'état et de CPU nécessaires aux hôtes d'extrémité. Seuls les fonctionnalités minimales nécessaires ont été incluses dans DCCP, laissant d'autres fonctionnalités, comme la correction d'erreurs directe (FEC), la semi fiabilité et les flux multiples, être mises en couches par dessus DCCP si désiré.

Différentes formes de contrôle d'encombrement conformes sont appropriées pour différentes applications. Par exemple, les jeux en ligne pourraient vouloir faire une utilisation rapide de toute bande passante disponible, tandis que les supports de flux en continu pourraient troquer cette réactivité contre des débits plus stables, moins saccadés. (Des variations brusques de débit peuvent provoquer des erreurs temporaires inacceptables, comme des pauses audibles ou des clics dans le flux

d'exécution.) DCCP permet donc aux applications de choisir parmi un ensemble de mécanismes de contrôle d'encombrement. Une solution de remplacement, du type du contrôle d'encombrement de TCP, diminue de moitié la fenêtre d'encombrement en réponse à un abandon ou marque de paquet, comme dans TCP. Les applications qui utilisent ce mécanisme de contrôle d'encombrement répondront rapidement aux changements de la bande passante disponible, mais elles doivent tolérer les changements brusques de la fenêtre d'encombrement typiques de TCP. Une seconde alternative, le contrôle en douceur de débit TCP (TRFC, *TCP-Rate Control Friendly*) [RFC3448], une forme de contrôle d'encombrement fondée sur une équation, minimise les changements brusques dans les taux d'envoi tout en préservant l'équité à long terme avec TCP. D'autres solutions de remplacement pourront être ajoutées lorsque de futurs mécanismes de contrôle d'encombrement seront standardisés.

DCCP permet également au trafic non fiable d'utiliser ECN en toute sécurité. Une interface de programmation d'application (API, *Application Programming Interface*) de noyau UDP pourrait ne pas permettre aux applications de régler les paquets UDP comme capables d'ECN, car l'API ne pourrait pas garantir que l'application va détecter ou répondre correctement à l'encombrement. Les API de noyau DCCP n'auront pas de tels problèmes, car DCCP met lui-même en œuvre le contrôle d'encombrement.

Nous avons choisi de ne pas exiger l'utilisation du gestionnaire d'encombrement [RFC3124], ce qui permet à plusieurs flux simultanés entre les mêmes envoyeurs et receveurs de partager le contrôle d'encombrement. Le gestionnaire d'encombrement (CM, *Congestion Manager*) actuel ne peut être utilisé par les applications qui ont leurs propres retours de bout en bout sur les pertes de paquets, mais ce n'est pas le cas pour beaucoup des applications qui utilisent actuellement UDP. De plus, le gestionnaire d'encombrement actuel ne prend pas facilement en charge des mécanismes multiples de contrôle d'encombrement ou des mécanismes où l'état des pertes ou marques de paquet passés est conservé chez le receveur plutôt que chez l'envoyeur. DCCP devrait être en mesure de faire usage du gestionnaire d'encombrement lorsque c'est demandé par l'application, mais on ne voit aucun avantage à faire dépendre le déploiement de DCCP de celui du CM lui-même.

L'intention, pour les mécanismes du protocole DCCP qui sont décrits dans le présent document, est de convenir à toute application qui désire des flux à encombrement contrôlé de datagrammes non fiables. Toutefois, les mécanismes de contrôle d'encombrement actuellement approuvés pour une utilisation avec DCCP, qui sont décrits dans les profils d'identifiant de contrôle d'encombrement séparés [RFC4341], [RFC4342], peuvent entraîner des problèmes pour certaines applications, y compris la vidéo interactive à forte largeur de bande. Ces applications devraient être capables d'utiliser DCCP une fois normalisés les profils d'identifiant de contrôle d'encombrement convenables.

3. Conventions et terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans ce document doivent être interprétés comme décrit dans la [RFC2119].

3.1 Nombres et champs

Toutes les quantités multi octets numériques dans DCCP, telles que les numéros d'accès, numéros de séquence, et les arguments pour les options, sont transmis dans l'ordre des octets du réseau (octet de poids fort en premier).

Nous faisons parfois référence aux côtés "gauche" et "droite" d'un champ de bits. "Gauche" signifie vers le bit de plus fort poids, et "droite" signifie vers le bit de moindre poids.

Les nombres aléatoires dans DCCP sont utilisés pour leurs propriétés de sécurité et devraient être choisis selon les directives de la [RFC4086].

Toutes les opérations sur les numéros de séquence DCCP utilisent l'arithmétique circulaire modulo 2^48, comme le font des comparaisons telles que "plus grand que" et "le plus grand". Cette forme de l'arithmétique conserve les relations entre les numéros de séquence lorsque ils reviennent de 2^48 - 1 à 0. Les stratégies de mise en œuvre pour les numéros de séquence DCCP ressembleront à celles des autres espaces d'arithmétique circulaire, y compris des numéros de séquence TCP [RFC793] et des numéros de série du DNS [RFC1982]. Il peut être judicieux de stocker les numéros de séquence DCCP dans les 48 bits de poids fort des entiers de 64 bits et mettre les 16 bits de moindre poids à zéro, puisque cela prend en charge une technique courante qui met en œuvre une comparaison circulaire A < B en vérifiant si (A - B) < 0 en utilisant l'arithmétique conventionnelle de complément à deux.

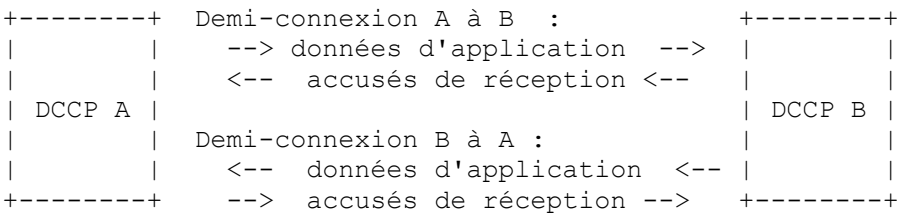
Les champs binaires réservés dans les en-têtes de paquets DCCP DOIVENT être mis à zéro par les envoyeurs et DOIVENT

être ignorés par les receveurs, sauf indication contraire. Cela permet des extensions futures du protocole. En particulier, les processeurs DCCP NE DOIVENT PAS réinitialiser une connexion DCCP simplement parce qu'un champ réservé a une valeur non nulle [RFC3360].

3.2. Parties d'une connexion

Chaque connexion DCCP fonctionne entre deux hôtes, qu'on appelle souvent DCCP A et DCCP B. Chaque connexion est initiée activement par l'un des hôtes, que nous appelons le client, l'autre, l'hôte d'abord passif, est appelé le serveur. Le terme "point d'extrémité DCCP" est utilisé pour désigner l'un ou l'autre des deux hôtes explicitement nommés par la connexion (le client et le serveur). Le terme "processeur DCCP" se réfère plus généralement à tout hôte qui pourrait avoir besoin de traiter un en-tête DCCP, ce qui comprend les points d'extrémité et tout boîtier de médiation sur le chemin, comme des pare-feu et des traducteurs d'adresses réseau.

Les connexions DCCP sont bidirectionnelles : les données peuvent passer de l'un ou l'autre des points d'extrémité à l'autre. Cela signifie que les données et les accusés de réception peuvent circuler dans les deux directions simultanément. Logiquement, toutefois, une connexion DCCP se compose de deux connexions unidirectionnelles séparées, appelées demi-connexions. Chaque demi-connexion consiste en les données d'application envoyées par une extrémité et les accusés de réception correspondants envoyés par l'autre extrémité. Nous pouvons illustrer cela comme suit :



Bien qu'elles soient logiquement distinctes, dans la pratique, les demi-connexions se chevauchent ; un paquet DCCP-DataAck, par exemple, contient les données d'application pertinentes pour une demi-connexion et les informations d'accusé de réception pertinentes pour l'autre.

Dans le contexte d'une seule demi-connexion, les termes "envoyeur HC" et "receveur HC" désignent, respectivement, l'extrémité d'envoi des données d'application et celle d'envoi des accusés de réception. Par exemple, DCCP A est l'envoyeur HC et DCCP B est le receveur HC dans la demi-connexion A à B.

3.3 Caractéristiques

Une caractéristique DCCP est un attribut de connexion sur la valeur duquel les deux terminaux s'accordent. Beaucoup de propriétés d'une connexion DCCP sont contrôlées par des caractéristiques, y compris les mécanismes de contrôle d'encombrement utilisés sur les deux demi-connexions. Les points d'extrémité réalisent un accord par le biais de l'échange d'options de négociation de caractéristiques dans les en-têtes DCCP.

Les caractéristiques DCCP sont identifiées par un numéro de caractéristique et un point d'extrémité. La notation "F / X" représente la caractéristique de numéro F située au point d'extrémité DCCP X. Chaque numéro de caractéristique valide correspond donc à deux caractéristiques, qui sont négociées séparément et n'ont pas besoin d'avoir la même valeur. Les deux points extrémités connaissent, et s'entendent sur, la valeur de chaque caractéristique valide. Le DCCP A est la "localisation de caractéristique" pour toutes les caractéristiques F / A, et la "caractéristique distante" pour toutes les caractéristiques F / B.

3.4. Temps d'aller-retour

Les mesures du temps d'aller-retour de DCCP sont effectuées par les mécanismes de contrôle d'encombrement ; différents mécanismes peuvent mesurer le temps d'aller-retour de façons différentes, ou ne pas le mesurer du tout. Cependant, le principal protocole DCCP utilise les temps d'aller-retour à l'occasion, comme dans les valeurs initiales de certains temporisateurs. Chaque mise en œuvre de DCCP définit donc un temps d'aller-retour par défaut à utiliser lorsque aucune estimation n'est disponible. Ce paramètre doit par défaut prendre une valeur de pas moins de 0,2 seconde, qui est un temps d'aller-retour raisonnablement conservateur pour les connexions TCP Internet. Le comportement de protocole spécifié en termes de "temps d'aller-retour" se réfère en fait à "une estimation du temps d'aller-retour actuel prise par un CCID quelconque, ou, si aucune estimation n'est disponible, la valeur de paramètre par défaut de temps d'aller-retour".

La durée de vie maximale d'un segment (MSL, *maximum segment lifetime*) est la longueur maximale du temps qu'un paquet

peut survivre dans le réseau. Le MSL DCCP devrait être égal à celui de TCP, qui est normalement de deux minutes.

3.5 Limitation de la sécurité

DCCP ne fournit aucune protection contre les agresseurs qui peuvent espionner sur une connexion en cours, ou qui peuvent deviner d'autres façons les numéros de séquence valides. Les applications désirant une sécurité renforcée devraient utiliser IPsec [RFC2401] ; selon le niveau de sécurité requis, une cryptographie de niveau application peut également suffire. Ces questions sont examinées au paragraphe 7.5.5 et à la Section 18.

3.6 Principe de robustesse

Les mises en œuvre de DCCP suivront le "principe général de robustesse" de TCP : "être conservateur dans ce que vous faites, être libéral dans ce que vous acceptez de la part des autres" [RFC0793].

4. Vue d'ensemble

La dynamique de haut niveau de la connexion DCCP fait écho à celle de TCP. Les connexions progressent en trois phases : l'initiation, dont une prise de contact en trois temps, le transfert de données, et la terminaison. Les données peuvent s'écouler dans les deux sens sur la connexion. Un cadre d'accusé de réception permet aux envoyeurs de découvrir combien de données ont été perdues et d'éviter ainsi d'encombrer à tort le réseau. Bien sûr, DCCP fournit une sémantique de datagramme non fiable, qui n'est pas la sémantique fiable du flux TCP. L'application doit conditionner ses données en trames explicites et doit retransmettre ses propres données si nécessaire. Il peut être utile de penser DCCP comme TCP moins la sémantique et la fiabilité du flux d'octets, ou comme UDP plus le contrôle d'encombrement, les prises de contact, et les accusés de réception.

4.1 Types de paquets

Dix types de paquets mettent en œuvre les fonctions du protocole DCCP. Par exemple, chaque nouvelle tentative de connexion commence par un paquet Demande-DCCP envoyée par le client. De cette façon, un paquet Demande-DCCP ressemble à un SYN TCP, mais comme Demande-DCCP est un type de paquet il n'y a aucun moyen d'envoyer une combinaison de fanions inattendue, telle que SYN + FIN + ACK + RST de TCP.

Huit types de paquets se produisent pendant le déroulement d'une connexion typique, montré ici. Noter les prises de contact en trois temps lors de l'initiation et la terminaison.

```
Client                                          Serveur
                        (1) Initiation
Demande-DCCP -->
                                                <-- Réponse-DCCP
DCCP-Ack     -->
                        (2) Transfert de données
DCCP-Data, DCCP-Ack, DCCP-DataAck -->
                        <-- DCCP-Data, DCCP-Ack, DCCP-DataAck
                         (3) Terminaison
                                                <-- DCCP-CloseReq
DCCP-Close     -->
                                                <-- DCCP-Reset
```

Les deux types de paquets restants sont utilisés pour resynchroniser après des salves de pertes.

Chaque paquet DCCP commence par un en-tête générique de taille fixe. Les types de paquets particuliers comprennent des données d'en-tête de taille fixe supplémentaires ; par exemple, les accusés de réception DCCP (DCCP-Ack) incluent un numéro d'accusé de réception. Les options DCCP et toutes les données des applications suivent l'en-tête de taille fixe.

Les types de paquets sont comme suit :
Demande-DCCP : Envoyé par le client pour établir une connexion (la première partie de la prise de contact en trois temps).
DCCP-Réponse : Envoyé par le serveur en réponse à une Demande-DCCP (la deuxième partie de la prise de contact en trois temps).
DCCP-Data : Utilisé pour transmettre des données d'application.

DCCP-Ack : Permet de transmettre les accusés de réception purs.
DCCP-DataAck : Utilisé pour transmettre des données d'application qui portent des informations d'accusé de réception.
DCCP-CloseReq : Envoyé par le serveur pour demander que le client ferme la connexion.
DCCP-Close : Utilisé par le client ou le serveur pour fermer la connexion; suscite une DCCP-Reset en réponse.
DCCP-Reset : Utilisé pour mettre fin à la connexion, normalement ou anormalement.
DCCP-Sync, DCCP-SyncAck : Utilisé pour resynchroniser les numéros de séquence, après une salve de pertes.

4.2 Séquençage de paquets

Chaque paquet DCCP porte un numéro de séquence de sorte que les pertes peuvent être détectées et signalées. Contrairement aux numéros de séquence TCP, qui sont fondés sur l'octet, les numéros de séquence DCCP s'incrémentent de un par paquet. Par exemple :

```
DCCP A                                          DCCP B
    DCCP-Data(seqno 1) -->
    DCCP-Data(seqno 2) -->
                                        <-- DCCP-Ack(seqno 10, ackno 2)
    DCCP-DataAck(seqno 3, ackno 10) -->
                                        <-- DCCP-Data(seqno 11)
```

Chaque paquet DCCP incrémente le numéro de séquence, qu'il contienne ou non des données d'application. Les accusés de réception DCCP-Ack purs incrémentent le numéro de séquence : par exemple, le deuxième paquet DCCP B ci-dessus utilise le numéro de séquence 11, car le numéro de séquence 10 a été utilisé pour un accusé de réception. Cela permet aux points d'extrémité de détecter toutes les pertes de paquets, y compris la perte de l'accusé de réception. Cela signifie aussi que les points d'extrémité peuvent devenir désynchronisés, après de longues rafales de perte. Les types de paquets DCCP-Sync et DCCP-SyncAck sont utilisés pour récupérer (paragraphe 7.5).

Comme DCCP fournit une sémantique non fiable, il n'y a pas de retransmission, et avoir un champ d'accusé de réception cumulatif de style TCP n'a pas de sens. Le champ Numéro d' accusé de réception de DCCP est égal au plus grand numéro de séquence reçu, plutôt qu'au plus petit numéro de séquence non reçu. Des options distinctes indiquent tout numéro de séquence intermédiaire qui n'a pas été reçu.

4.3 États

Les points d'extrémité DCCP progressent à travers différents états au cours d'une connexion, correspondant en gros aux trois phases de l'initiation, du transfert de données, et de terminaison. La figure ci-dessous montre la progression typique à travers ces états pour un client et un serveur.

```
Client                                                  Serveur
                        (0) pas de connexion
CLOSED                                                  LISTEN
                        (1) Initiation
REQUEST     Demande-DCCP -->
                                        <-- DCCP-Réponse    RESPOND
PARTOPEN    DCCP-Ack ou DCCP-DataAck -->
                        (2) Transfert de données
OPEN        <-- DCCP-Data, Ack, DataAck -->                 OPEN
                        (3) Terminaison
                                        <-- DCCP-CloseReq   CLOSEREQ
CLOSING     DCCP-Close -->
                                        <-- DCCP-Reset      CLOSED
TIMEWAIT
CLOSED
```

Les neuf états possibles sont les suivants. Ils sont énumérés en ordre croissant, de sorte que "l'état >= CLOSEREQ" signifie la même chose que "état = CLOSEREQ" ou "état = CLOSING" ou "état = TIMEWAIT". La Section 8 décrit les états de façon plus détaillée.

FERMÉ (CLOSED) : Représente les connexions inexistantes.

ÉCOUTER (LISTEN) : Représente les prises de serveur dans l'état passif à l'écoute. ÉCOUTER et FERMÉ ne sont

associés à aucune connexion DCCP particulière.

DEMANDE (REQUEST) : Une prise de client entre dans cet état, à partir de FERMÉ, après envoi d'un paquet Demande-DCCP pour essayer d'initier une connexion.

RÉPONDRE (RESPOND) : Une prise de serveur entre dans cet état, à partir de ÉCOUTER, après avoir reçu une Demande-DCCP d'un client.

PARTOPEN : Une prise de client entre dans cet état, à partir de DEMANDE, après avoir reçu une DCCP-RÉPONSE du serveur. Cet état constitue la troisième phase de la prise de contact à trois temps. Le client peut envoyer les données d'application dans cet état, mais il DOIT comprendre un numéro d'accusé de réception sur tous ses paquets.

OUVERT (OPEN) : La partie centrale du transfert de données d'une connexion DCCP. Les prises de client et de serveur entrent dans cet état à partir de PARTOPEN et de RÉPONDRE, respectivement. Parfois, on parle d'états SERVEUR-OUVERT et CLIENT-OUVERT, ce qui correspond à l'état OUVERT du serveur et à l'état OUVERT du client.

CLOSEREQ (DEMANDE DE CLÔTURE) : Une prise de serveur entre dans cet état, à partir de SERVEUR-OUVERT, pour dire au client de fermer la connexion et de garder l'état TIMEWAIT.

CLÔTURE : Les prises serveur et client peuvent entrer dans cet état pour fermer la connexion.

TIMEWAIT : Une prise serveur ou client reste dans cet état pendant 2MSL (4 minutes) après que la connexion a été supprimée, afin de prévenir les erreurs dues à la livraison de vieux paquets. Seul l'un des points d'extrémité doit entrer dans l'état TIMEWAIT (l'autre peut entrer immédiatement dans l'état FERMÉ) et un serveur peut demander à son client de garder l'état TIMEWAIT en utilisant le type de paquet DCCP-CloseReq.

4.4 Mécanismes de contrôle d’encombrement

L’encombrement des connexions DCCP est contrôlé, mais contrairement à TCP, les applications DCCP ont le choix du mécanisme de contrôle d’encombrement. En fait, les deux demi-connexions peuvent être régies par des mécanismes différents. Les mécanismes sont notés par un octet d'identifiant de contrôle d’encombrement (CCID, *Control Congestion Identifer*). Les points d'extrémité négocient leurs CCID lors de l'initialisation de la connexion. Chaque CCID décrit comment l'envoyeur HC limite le débit de paquets de données, comment le receveur HC envoie les retours sur l'encombrement par les accusés de réception, etc.. Les CCID 2 et 3 sont actuellement définis ; les CCID 0, 1 et 4 à 255 sont réservés. D'autres CCID pourront être définis à l'avenir.

Le CCID 2 fournit un contrôle d'encombrement de type TCP, qui est similaire à celui de TCP. L'envoyeur tient une fenêtre d’encombrement et envoie les paquets jusqu'à ce que la fenêtre soit pleine. Le receveur accuse réception des paquets. Les paquets éliminés et les ECN [RFC3168] indiquent l'encombrement ; la réponse à l’encombrement est de réduire de moitié la fenêtre d’encombrement. Les accusés de réception dans CCID 2 contiennent les numéros de séquence de tous les paquets reçus dans une certaine fenêtre, semblable à un accusé de réception sélectif (SACK) [RFC2018].

Le CCID 3 fournit le contrôle de débit en douceur pour TCP (TFRC, *TCP-Friendly Rate Control*) une forme de contrôle d’encombrement fondée sur une équation qui vise à répondre à l’encombrement plus en douceur qu'avec CCID 2. L'envoyeur conserve un taux de transmission, qu'il met à jour en utilisant l'estimation du taux de perte et de marquage de paquets du receveur. Le CCID 3 se comporte un peu différemment que TCP dans le court terme, mais est conçu pour fonctionner de façon équitable avec TCP sur le long terme.

La Section 10 décrit les CCID de DCCP plus en détail. Les comportements des CCID 2 et 3 sont entièrement définis dans les documents de profil [RFC4341], [RFC4342].

4.5 Options de négociation de caractéristique

Les points d'extrémité DCCP utilisent les options Change et Confirme pour négocier et de s'entendre sur les valeurs de caractéristiques. La négociation de caractéristiques va presque toujours se produire sur la prise de contact d'initiation de connexion, mais elle peut commencer à tout moment.

Il y a en tout quatre options de négociation de caractéristique : Change L, Confirme L, Change R, et Confirme R. Les options "L" sont envoyés par la localisation de caractéristique et les options "R" sont envoyés par la caractéristique distante. Une option Change R indique à la localisation de caractéristique "modifier la valeur de cette caractéristique comme suit". La localisation de caractéristique répond avec Confirme L, qui signifie "je l'ai changée". Certaines caractéristiques

permettent que les options Change R contiennent plusieurs valeurs triées par ordre de préférence. Par exemple :

```
   Client                                              Serveur
   Change R(CCID, 2) -->
                                          <------- Confirme L(CCID, 2)
                              * accord pour que CCID/Serveur = 2 *
   Change R(CCID, 3 4) -->
                                          <-- Confirme L(CCID, 4, 4 2)
                              * accord pour que CCID/Serveur  = 4 *
```

Les deux échanges négocient la valeur de la caractéristique CCID/serveur, qui est le CCID en usage sur la demi-connexion serveur-client. Dans le deuxième échange, le client demande que le serveur utilise CCID 3 ou 4, 3 étant préféré, le serveur choisit 4 et fournit sa liste de préférence, "4 2".

Les options Change L et Confirme R sont utilisées pour des négociations de caractéristiques initiées par la localisation de caractéristique. Dans l'exemple suivant, le serveur demande que le CCID/Serveur soit réglé à 3 ou 2, avec 3 préféré, et le client l'accepte.

```
   Client                                              Serveur
                                          <-------- Change L(CCID, 3 2)
   Confirme R(CCID, 3, 3 2)  -->
                              * accord pour que CCID/Serveur = 3 *
```

La Section 6 décrit plus en détails les options de négociation de caractéristiques, y compris les stratégies de retransmission qui rendent la négociation fiable.

4.6 Différences avec TCP

Les différences de DCCP avec TCP en dehors de celles évoquées jusqu'ici comprennent les éléments suivants :

- Un espace copieux pour les options (jusqu'à 1008 octets ou la PMTU).

- Différents formats d'accusé de réception. Le CCID pour une connexion détermine la quantité d'informations d'accusé de réception qui doit être transmise. Par exemple, dans CCID 2 (de style TCP) c'est environ un accusé de réception pour deux paquets, et chaque accusé de réception doit déclarer exactement quels paquets ont été reçus. Dans CCID 3 (TFRC), il est d'environ un accusé de réception par temps d'aller-retour, et les accusés de réception doivent déclarer au minimum juste les longueurs des intervalles de pertes récentes.

- Protection contre le déni de service (DoS). Plusieurs mécanismes aident à limiter la quantité d'état que des clients se conduisant éventuellement mal peuvent forcer les serveurs DCCP à entretenir. Une option Init Cookie analogue aux cookies SYN de TCP [SYNCOOKIES] évite des attaques du style inondation de SYN. Un seul point d'extrémité de la connexion doit garder l'état TIMEWAIT ; le paquet DCCP-CloseReq, qui ne peut être envoyé que par le serveur, passe cet état au client. Diverses limites de débit permettent aux serveurs d'éviter des attaques qui pourraient forcer à des calculs ou une génération de paquets extensifs.

- Distinguer les différents types de perte. Une option Données éliminées (paragraphe 11.7) permet à un point d'extrémité de déclarer qu'un paquet a été abandonné à cause de la corruption, en raison d'un débordement de mémoire tampon de réception, et ainsi de suite. Cela facilite la recherche sur les réponses plus appropriées de taux de contrôle à ces pertes non liées à l'encombrement du réseau (bien qu'actuellement de telles pertes provoquent une réponse d'encombrement).

- Capacité à fournir des accusés de réception. Dans TCP, un paquet ne peut être acquitté que une fois que les données sont mises en file d'attente de manière fiable en attente de la livraison à l'application. Cela n'a pas de sens dans DCCP, où une application peut, par exemple, demander un abandon du la partie frontale de la mémoire tampon de réception. Un paquet DCCP peut être acquitté dès que son en-tête a été correctement traité. Concrètement, un paquet devient acquittable à l'étape 8 du pseudo-code de traitement des paquets du paragraphe 8.5. La capacité à accuser réception ne garantit pas la livraison de données, cependant, l'option Données éliminées peut par la suite signaler que les données d'application du paquet ont été éliminées.

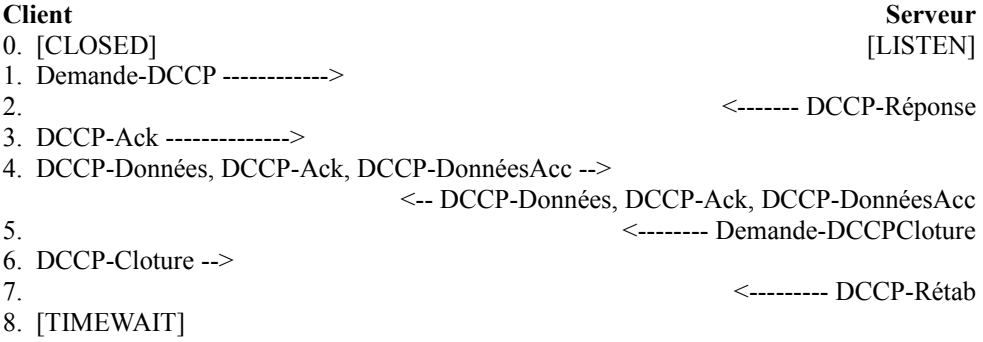
- Pas de fenêtre de réception. DCCP est un protocole de contrôle d'encombrement, pas un protocole de contrôle de flux.

- Aucune ouverture simultanée. Chaque connexion a un client et un serveur.

o Pas d'états mi-clos. DCCP n'a pas d'état correspondant au FINWAIT et CLOSEWAIT de TCP, où une demi-connexion est explicitement fermée tandis que l'autre est toujours active. Le code d'abandon 1 de l'option Données éliminées, Application non écoutante (paragraphe 11.7), peut cependant atteindre un effet similaire.

4.7 Exemple de connexion

La progression d'une connexion typique DCCP est comme suit. (Cette description est informative, non normative.)



1. Le client envoie au serveur un paquet Demande-DCCP précisant les accès client et serveur, le service demandé, et toutes les caractéristiques en cours de négociation, y compris le CCID que le client souhaite que le serveur utilise. Le client peut éventuellement faire porter une demande d'application sur le paquet Demande-DCCP. Le serveur peut ignorer cette demande d'application.

2. Le serveur envoie au client un paquet DCCP-Réponse indiquant qu'il est disposé à communiquer avec le client. Cette réponse indique toutes les caractéristiques et les options que le serveur accepte, entame la négociation d'autres caractéristique comme désiré, et éventuellement comprend les Cookies Init qui enveloppent toutes ces informations et qui doivent être retournés par le client pour que la connexion soit complète.

3. Le client envoie au serveur un paquet DCCP-Ack qui accuse réception du paquet DCCP-Réponse. Cela accuse réception du numéro de séquence initial du serveur et retourne tous les Init Cookies dans la DCCP-Réponse. Il peut également poursuivre les négociations de caractéristiques. Le client peut faire porter une demande de niveau application sur cet accusé de réception, produisant un paquet DCCP-DataAck.

4. Le serveur et le client échangent alors des paquets de données DCCP, des paquets DCCP-Ack qui accusent réception de ces données, et, éventuellement, des paquets DCCP-DataAck contenant des données avec les accusés de réception portés. Si le client n'a pas de données à envoyer, le serveur enverra alors des paquets DCCP-Data et DCCP-DataAck, tandis que le client enverra exclusivement des DCCP-Ack. (Toutefois, le client ne peut envoyer de paquets de données DCCP avant de recevoir au moins un paquet qui ne soit pas de réponse DCCP de la part du serveur.)

5. Le serveur envoie un paquet DCCP-CloseReq demandant la clôture.

6. Le client envoie un paquet DCCP-Close reconnaissant la clôture.

7. Le serveur envoie un paquet DCCP-Reset avec le code Réinitialiser 1, "Fermé", et efface son état de connexion. Les DCCP-Reset font partie de la terminaison normale de la connexion, voir au paragraphe 5.6.

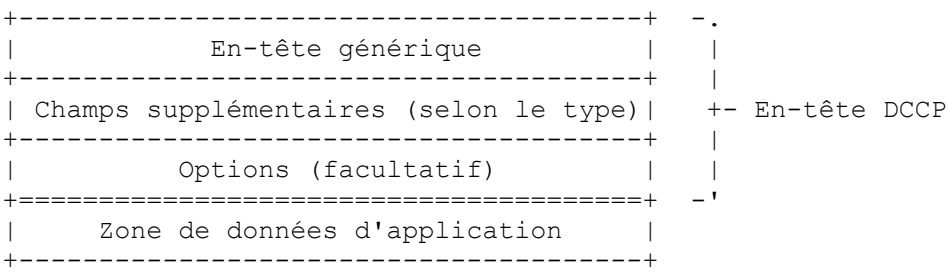
8. Le client reçoit le paquet DCCP-Reset et conserve l'état pour deux segments de durée de vie maximale (MSL, *Maximum Segment Lifetime*) pour permettre à tous les paquets restants de quitter le réseau.

Une autre séquence de clôture de connexion est initiée par le client :
5b. Le client envoie un paquet DCCP-Fermer qui ferme la connexion.
6b. Le serveur envoie un paquet DCCP-Reset avec le code Réinitialiser 1, "Fermé", et efface son état de connexion.
7b. Le client reçoit le paquet DCCP-Reset et garde l'état pour 2 MSL pour permettre à tous les paquets restants de quitter le réseau.

5. Formats de paquet

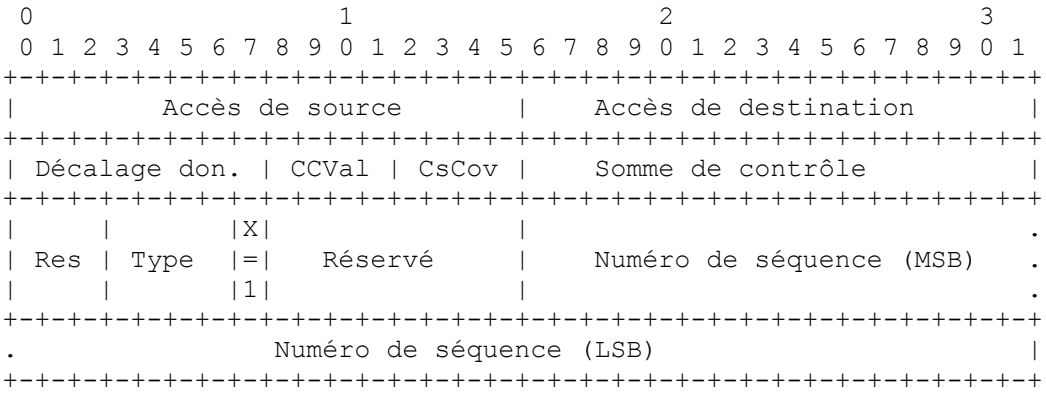
L'en-tête DCCP peut être long de 12 à 1020 octets. La partie initiale de l'en-tête a la même sémantique pour tous les types

de paquets actuellement définis. Après cela viennent tous les champs de longueur fixe requis par le type de paquet, puis une liste de longueur variable d'options. La zone de données d'application suit l'en-tête. Dans certains types de paquets, cette zone contient des données pour l'application ; dans d'autres types de paquets, son contenu est ignoré.

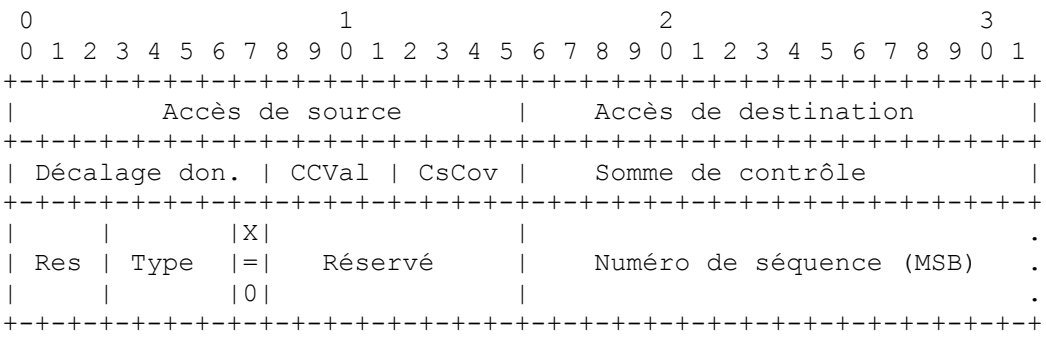


5.1 En-tête générique

L'en-tête DCCP générique prend des formes différentes selon la valeur de X, le bit de numéro de séquence étendu. Si X est un, le champ Numéro de séquence est long de 48 bits, et l'en-tête générique prend 16 octets, comme suit.



Si X est zéro, seuls les 24 bits de moindre poids (LSB, *least significant bits*) du numéro de séquence sont transmis, et l'en-tête générique est long de 12 octets.



Les champs d'en-tête générique sont définis comme suit.

Accès de source et de destination : 16 bits chacun
Ces champs identifient la connexion, comme les champs correspondants de TCP et UDP. L'accès de source représente l'accès pertinent sur le point d'extrémité qui a envoyé ce paquet, et l'accès de destination représente l'accès pertinent sur l'autre point d'extrémité. Lors de l'initiation d'une connexion, le client DEVRAIT choisir son accès de source aléatoirement pour réduire la probabilité d'une attaque.

Les API DCCP devraient traiter les numéros d'accès de la même manière que les numéros d'accès TCP et UDP. Par exemple, les machines qui distinguent les accès "privilégiés" et "non privilégiés" pour TCP et UDP devraient faire de même pour DCCP.

Décalage des données : 8 bits
Le décalage depuis le début de l'en-tête du paquet DCCP au début de la zone des données d'application, en mots de 32 bits. Le receveur DOIT ignorer les paquets dont le décalage des données est inférieur à la taille minimale d'en-tête pour le type

donné ou supérieur au paquet DCCP lui-même.

CCVal : 4 bits
Utilisé par le CCID de la demi-connexion envoyeur. Par exemple, l'envoyeur du CCID de A à B, qui est actif au DCCP A, PEUT envoyer 4 bits d'information par paquet à son receveur en codant cette information dans CCVal. L'envoyeur DOIT mettre CCVal à zéro à moins que son CCID de demi-connexion envoyeur n'en dispose autrement, et le receveur DOIT ignorer le champ CCVal à moins que son CCID de demi-connexion receveur spécifie le contraire.

Couverture de somme de contrôle (CsCov) : 4 bits
Couverture de somme de contrôle détermine les parties du paquet qui sont couvertes par le champ Somme de contrôle. Cela inclut toujours l'en-tête DCCP et les options, mais certaines, ou la totalité, des données d'application peuvent être exclues. Cela peut améliorer les performances sur des liaisons bruyantes pour les applications qui peuvent tolérer la corruption. Voir la Section 9.

Somme de contrôle : 16 bits
La somme de contrôle Internet de l'en-tête du paquet DCCP (y compris les options), un pseudo en-tête de couche réseau, et, selon la couverture de somme de contrôle, toutes, certaines, ou aucune des données d'application. Voir la Section 9.

Réservé (Res) : 3 bits. Les envoyeurs DOIVENT régler ce champ tout à zéro sur les paquets générés, et les receveurs DOIVENT ignorer sa valeur.

Type : 4 bits. Le champ Type indique le type du paquet. Les valeurs définies suivent :

Type	**Signification**
0	Demande-DCCP
1	DCCP-Réponse
2	DCCP-Données
3	DCCP-Ack
4	DCCP-DataAck
5	DCCP-CloseReq
6	DCCP-Close
7	DCCP-Reset
8	DCCP-Sync
9	DCCP-SyncAck
10-15	réservés

Tableau 1 : Types de paquets DCCP

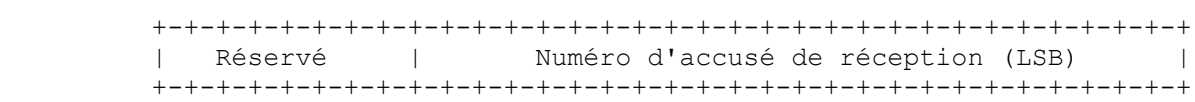
Les receveurs DOIVENT ignorer tous les paquets de type réservé. C'est-à-dire, les paquets de type réservé NE DOIVENT PAS être traités, et ils NE DOIVENT PAS être acquittés comme reçus.

Numéros de séquence étendus (X) : 1 bit
Réglé à un pour indiquer l'utilisation d'un en-tête générique étendu avec des numéros de séquence et d'accusé de réception de 48 bits. Les paquets DCCP-Données, DCCP-DataAck et DCCP-Ack PEUVENT régler X à zéro ou un. Tous les paquets Demande-DCCP, DCCP-Réponse, DCCP-CloseReq, DCCP-Close, DCCP-Reset, DCCP-Sync, et DCCP-SyncAck DOIVENT régler X à un ; les points d'extrémité DOIVENT ignorer tous les paquets avec X mis à zéro. Les connexions à haut débit DEVRAIENT régler X à un sur tous les paquets pour une protection accrue contre les attaques d'enveloppement de numéros de séquence. Voir le paragraphe 7.6.

Numéro de séquence : 48 ou 24 bits
Identifie le paquet de façon univoque dans la séquence de tous les paquets envoyés sur cette connexion par la source. Le numéro de séquence augmente de un à chaque paquet envoyé, y compris les paquets tels que DCCP-Ack qui ne portent pas de données d'application. Voir la Section 7.

Tous les types de paquets définis actuellement, sauf Demande-DCCP et DCCP-Données portent un sous en-tête de numéro d'accusé de réception dans les quatre ou huit octets immédiatement après l'en-tête générique. Quand X = 1, son format est le suivant :

Quand X = 0, seuls les 24 bits de moindre poids du numéro d'accusé de réception sont transmis, donnant le format suivant au sous en-tête de numéro d'accusé de réception :



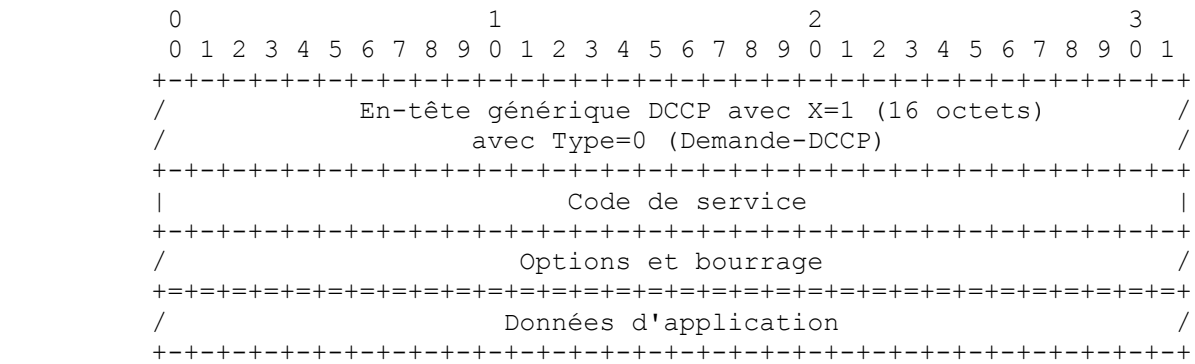
Réservé : 16 ou 8 bits
Les envoyeurs DOIVENT régler ce champ tout à zéro sur les paquets générés, et le receveur DOIT ignorer sa valeur.

Numéro d'accusé de réception : 48 ou 24 bits
Contient généralement le plus grand numéro de séquence reçu (GSR, *Greatest Sequence number Received*) sur tout paquet acquittable jusqu'ici. Un paquet est acquittable si et seulement si son en-tête a été correctement traité par le receveur ; ceci est décrit plus en détails au paragraphe 7.4. Des options telles que Vecteur Ack (paragraphe 11.4) se combinent avec le numéro d'accusé de réception pour fournir des informations précises sur les paquets qui sont arrivés.

Les numéros d'accusé de réception sur les paquets DCCP-Sync et DCCP-SyncAck n'ont pas besoin d'être égaux au GSR. Voir le paragraphe 5.7.

5.2 Paquets Demande-DCCP

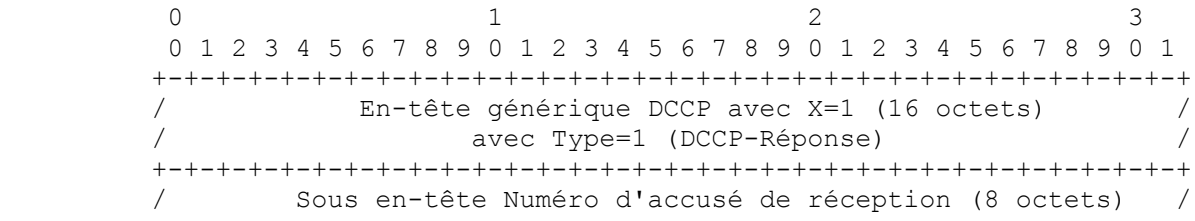
Un client établit une connexion DCCP en envoyant un paquet Demande-DCCP. Ces paquets PEUVENT contenir des données d'application et DOIVENT utiliser des numéros de séquence de 48 bits (X = 1).

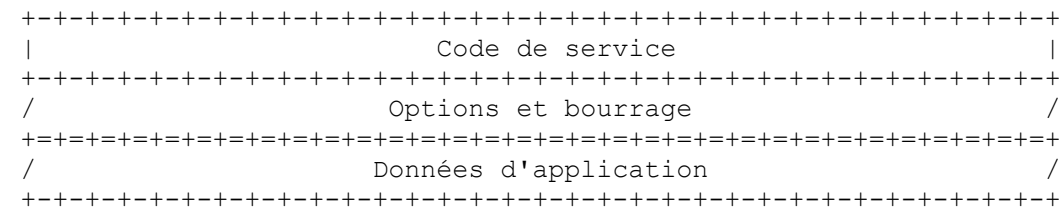


Code de service : 32 bits
Décrit le service au niveau applicatif pour lequel l'application cliente veut se connecter. Les codes de service sont destinés à fournir des informations sur le protocole d'application qu'une connexion a l'intention d'utiliser, aidant ainsi les boîtiers de médiation et réduisant la dépendance à des accès mondialement bien connus. Voir le paragraphe 8.1.2.

5.3 Paquets DCCP-Réponse

Le serveur répond aux paquets Demande-DCCP par des paquets DCCP-Réponse. Il s'agit de la deuxième phase de la prise de contact à trois temps. Les paquets DCCP-Réponse peuvent contenir des données d'application et DOIVENT utiliser des numéros de séquence de 48 bits (X = 1).



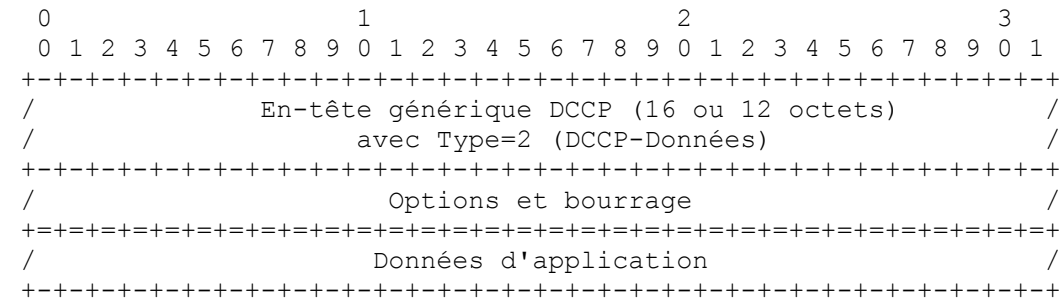


Numéro d'accusé de réception : 48 bits
Contient le GSR. Comme les DCCP-Réponse ne sont envoyés que lors de initialisation de la connexion, ce sera toujours égal au numéro de séquence sur une Demande-DCCP reçue.

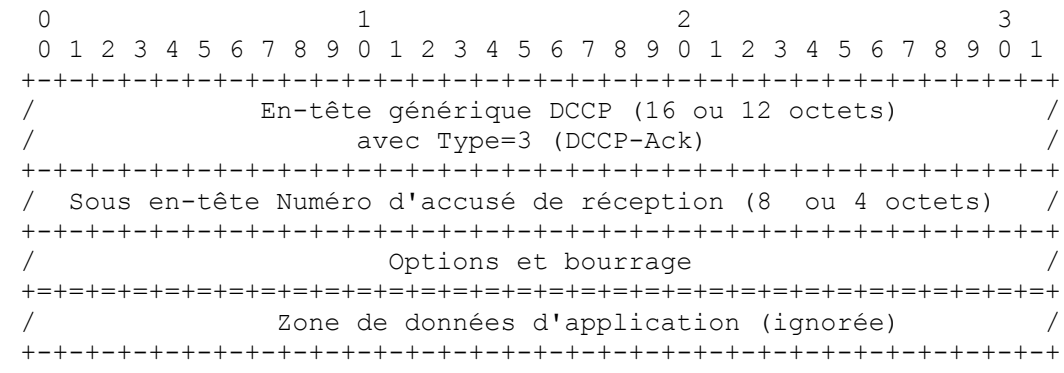
Code de service : 32 bits. DOIT être égal au code de service de la Demande-DCCP correspondante.

5.4 Paquets DCCP-Data, DCCP-Ack et DCCP-DataAck

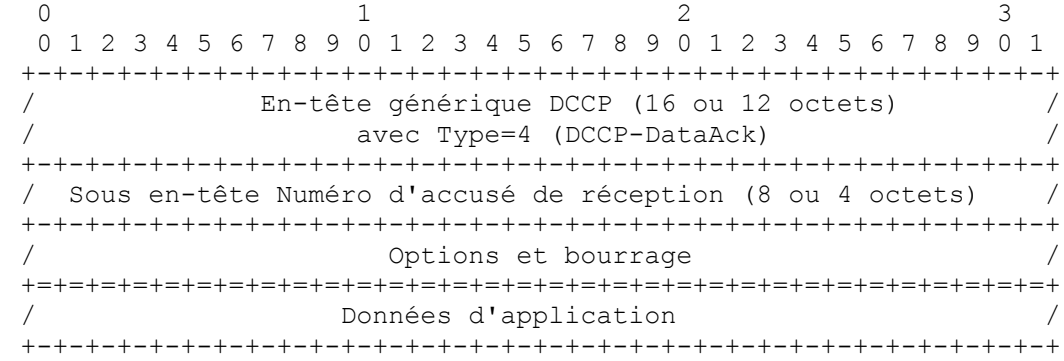
La partie transfert de données centrale de chaque connexion utilise des paquets DCCP DCCP-Data, DCCP-Ack et DCCP-DataAck. Ces paquets PEUVENT utiliser des numéros de séquence de 24 bits, selon la valeur de l'option de caractéristique Autoriser les numéros de séquence courts (paragraphe 7.6.1). Les paquets DCCP-Données portent des données d'application, sans accusé de réception.



Les paquets DCCP-Ack ne portent pas les données, mais contiennent un numéro d'accusé de réception. Ils sont utilisés pour de purs accusés de réception.



Les paquets DCCP-DataAck transportent des données d'applications et un numéro d'accusé de réception. Ils portent les informations d’accusé de réception sur un paquet de données.



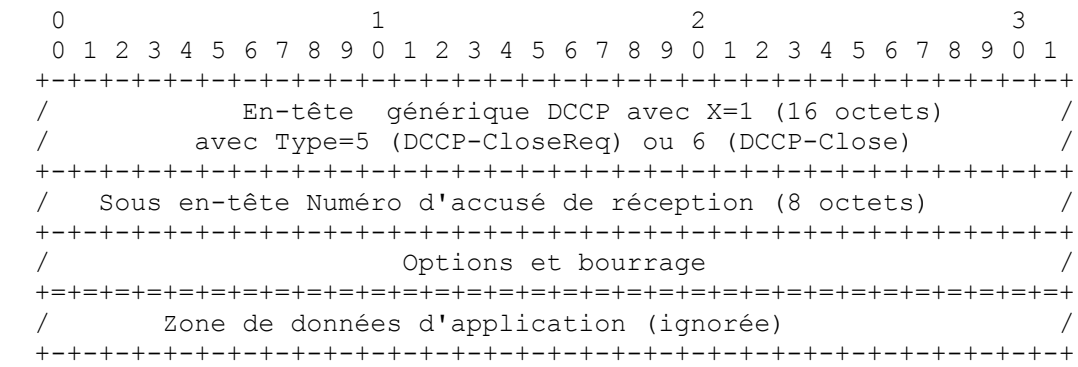
Un paquet DCCP-Data ou DCCP-DataAck peut avoir une zone de données d'application de longueur nulle, ce qui indique que l'application a envoyé un datagramme de longueur nulle. Cela diffère des paquets Demande-DCCP et DCCP-Réponse, où une zone de données d'application vide indique l'absence de données d'application (et non la présence de données d'application de longueur nulle). L'API DEVRAIT signaler tous datagrammes reçus de longueur nulle à l'application receveuse.

Un paquet DCCP-Ack PEUT avoir une zone de données d'application de longueur non nulle, qui bourre essentiellement les DCCP-Ack à la longueur souhaitée. Les receveurs DOIVENT ignorer le contenu de la zone de données d'application dans les paquets DCCP-Ack.

Les paquets DCCP-Ack et DCCP-DataAck incluent souvent d'autres options que d'accusé de réception, telles que Vecteur Ack, comme requis par le mécanisme de contrôle d'encombrement en usage.

5.5 Paquets DCCP-CloseReq et DCCP-Close

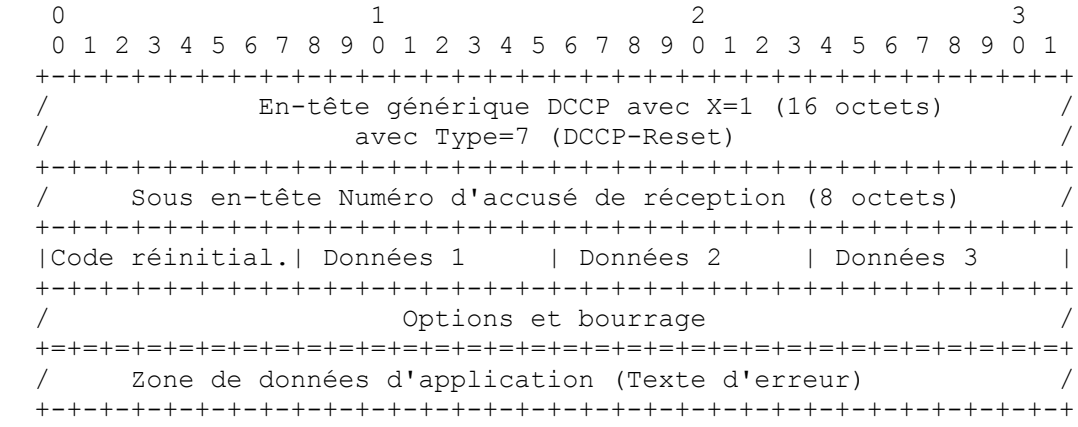
Les paquets DCCP-CloseReq et DCCP-Close commencent la prise de contact qui termine normalement une connexion. Le client ou le serveur peut envoyer un paquet DCCP-Close, ce qui permettra d'obtenir un paquet DCCP-Reset. Seul le serveur peut envoyer un paquet DCCP-CloseReq, ce qui indique que le serveur veut fermer la connexion, mais ne veut pas garder son état TIMEWAIT. Les deux types de paquets DOIVENT utiliser des numéros de séquence de 48 bits (X = 1).



Comme avec les paquets DCCP-Ack, DCCP-CloseReq et DCCP-Close, les paquets PEUVENT avoir des zones de données d'application de longueur non nulle , dont les receveurs DOIVENT ignorer le contenu.

5.6 Paquets DCCP-Reset

Les paquets DCCP-Reset ferment inconditionnellement une connexion. Les connexions se terminent normalement par un DCCP-Reset, mais les réinitialisations peuvent être envoyées pour d'autres raisons, y compris de mauvais numéros d'accès, un mauvais comportement d'option, des échos de nom occasionnel ECN incorrects, et ainsi de suite. Les DCCP-Reset DOIVENT utiliser des numéros de séquence de 48 bits (X = 1).



Code de réinitialisation : 8 bits
Représente la raison pour laquelle l'envoyeur réinitialise la connexion DCCP.

Données 1, Données 2 et Données 3 : 8 bits chacun
Les champs Données fournissent des informations supplémentaires sur les raisons pour lesquelles l'envoyeur réinitialise la connexion DCCP. La signification de ces champs dépend de la valeur du code de réinitialisation.

Zone de données d'application : Texte d'erreur
S'il est présent, le texte d'erreur est une chaîne de texte lisible par l'homme codée en Unicode UTF-8, et de préférence en anglais, qui décrit l'erreur plus en détail. Par exemple, un DCCP-Reset avec le code de réinitialisation 11, "Pénalité d'agression", pourrait contenir un texte d'erreur tel que "Pénalité d'agression : reçu trois mauvais échos de nom occasionnel ECN, faisant supposant un mauvais comportement".

Les codes de réinitialisation suivants sont actuellement définis. Sauf spécification contraire, les champs Données 1, 2 et 3 DOIVENT être mis à 0 par l'envoyeur du DCCP-Reset et ignorés par leur receveur. le paragraphe Références décrit des situations concrètes qui causent la génération de chaque code de réinitialisation ; la liste n'est pas exhaustive.

0, "non spécifié"
Indique l'absence d'un code de réinitialisation significatif. L'utilisation du code de réinitialisation 0 n'est pas recommandée : l'envoyeur devrait choisir un code de réinitialisation qui définit plus clairement pourquoi la connexion est en cours de réinitialisation.

1, "Fermé"
Fermeture normale de la connexion. Voir le paragraphe 8.3.

2, "Interrompue"
Le point d'extrémité d'envoi a renoncé à envoyer sur la connexion en raison d'un manque de progrès. Voir les paragraphes 8.1.1 et 8.1.5.

3, "Pas de connexion"
Aucune connexion n'existe. Voir le paragraphe 8.3.1.

4, "Erreur de paquet"
Un paquet valide est arrivé avec un type inattendu. Par exemple, un paquet DCCP-Données avec une somme de contrôle d'en-tête et des numéros de séquence valide est arrivé à une connexion dans l'état DEMANDE. Voir le paragraphe 8.3.1. Le champ Données 1 est égal au type de paquet fautif comme un nombre de huit bits : ainsi, un paquet fautif de type 2 se traduira par une valeur de Données 1 de 2.

5, "Erreur d'option"
Une option était erronée, et l'erreur était suffisamment grave pour réclamer la réinitialisation de la connexion. Voir les paragraphes 6.6.7, 6.6.8, et 11.4. Le champ Données 1 est égal au type d'option fautif ; Données 2 et Données 3 sont égaux aux deux premiers octets des données d'option (ou zéro si l'option avait moins de deux octets de données).

6, "Erreur obligatoire"
Le point d'extrémité d'envoi n'a pas pu traiter une option 0 qui était immédiatement précédé de Obligatoire. Les champs Données rapportent le type d'option et les données de l'option 0, en utilisant le format de code de réinitialisation 5, "Erreur d'option". Voir le paragraphe 5.8.2.

7, "Connexion refusée"
L'accès de destination ne correspond pas à un accès ouvert pour l'écoute. Envoyé uniquement en réponse aux Demande-DCCP. Voir le paragraphe 8.1.3.

8, "Mauvais code de service"
Le code de service n'est pas égal au code du service attaché à l'accès de destination. Envoyé uniquement en réponse aux Demande-DCCP. Voir le paragraphe 8.1.3.

9, "Trop occupé"
Le serveur est trop occupé pour accepter de nouvelles connexions. Envoyé uniquement en réponse aux Demande-DCCP. Voir le paragraphe 8.1.3.

10, "Mauvais cookie Init"
Le cookie Init repris par le client était incorrect ou manquant. Voir le paragraphe 8.1.4.

11, "Pénalité d'agression"
Ce point d'extrémité a détecté un mauvais comportement en relation avec le contrôle d’encombrement de la part de l'autre extrémité. Voir le paragraphe 12.3.

12-127, réservés
Les receveurs devraient traiter ces codes comme ils le font du code de réinitialisation 0, "non spécifié".

128-255, codes spécifiques CCID
Leur sémantique dépend du CCID de la connexion. Voir le paragraphe 10.3. Les receveurs devraient traiter les codes de réinitialisation spécifiques de CCID inconnus comme ils le font du code de réinitialisation 0, "non spécifié".

Le tableau suivant résume ces informations.

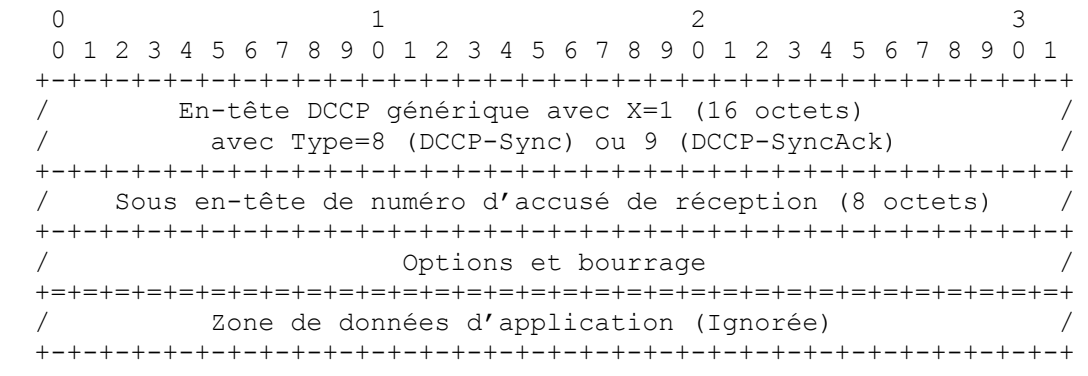
Code de réinitialisation	Nom	Données 1	Données 2 & 3
0	Non spécifié	0	0
1	Fermé	0	0
2	Interrompu	0	0
3	Pas de connexion	0	0
4	Erreur de paquet	Type de paquet	0
5	Erreur d'option	n° d'option	données d'option
6	Erreur obligatoire	n° d'option	données d'option
7	Connexion refusée	0	0
8	Mauvais code de service	0	0
9	Trop occupé	0	0
10	Mauvais cookie Init	0	0
11	Pénalité d'agression	0	0
12 à 127	Réservé		
128 à 255	Codes spécifiques de CCID		

Tableau 2 : Codes de réinitialisation DCCP

Les options sur les paquets DCCP-Reset sont traitées avant que la connexion soit fermée. Cela signifie que certaines combinaisons d'options, en particulier celles qui impliquent Obligatoire, peuvent entraîner un point d'extrémité à répondre à un DCCP-Reset valide par un autre DCCP-Reset. Cela ne peut pas conduire à une tempête de réinitialisations ; comme le premier point d'extrémité a déjà réinitialisé la connexion, le deuxième DCCP-Reset sera ignoré.

5.7 Paquets DCCP-Sync et DCCP-SyncAck

Les paquets DCCP-Sync aident les points d'extrémité DCCP à récupérer la synchronisation après des rafales de pertes et de récupérer de connexions semi-ouvertes. Chaque DCCP-Sync valide reçu provoque immédiatement un DCCP-SyncAck. Les deux types de paquets DOIVENT utiliser des numéros de séquence de 48 bits (X = 1).



Le champ Numéro d'accusé de réception a une sémantique particulière pour les paquets DCCP-Sync et DCCP-SyncAck. Tout d'abord, le paquet correspondant à un numéro d'accusé de réception de DCCP-Sync n'a pas besoin d'avoir été acquittable. Donc, les receveurs NE DOIVENT PAS supposer qu'un paquet a été traité simplement parce qu'il apparaît dans le champ Numéro d'accusé de réception d'un paquet de DCCP-Sync. Cela diffère de tous les autres types de paquet, où le numéro d'accusé de réception correspond par définition à un paquet acquittable. Deuxièmement, le numéro d'accusé de réception sur tout paquet DCCP-SyncAck DOIT correspondre au numéro de séquence sur un paquet DCCP-Sync acquittable. En présence d'un déclassement, il pourrait ne pas être égal au GSR.

Comme avec les paquets DCCP-Ack, les paquets DCCP-Sync et DCCP-SyncAck PEUVENT avoir des zones de données d'application de longueur non nulle, que le receveur du contenu DOIT ignorer. Les paquets DCCP-Sync bourrés peuvent

être utiles lors de l'exécution de la découverte de la MTU du chemin ; voir la Section 14.

5.8 Options

Tout paquet DCCP peut contenir des options, qui occupent l'espace à la fin de l'en-tête DCCP. Chaque option est un multiple de 8 bits. Les options individuelles ne sont pas bourrées à des multiples de 32 bits, et toute option peut commencer sur n'importe quelle limite d'octet. Cependant, la combinaison de toutes les options DOIT s'ajuster à un multiple de 32 bits ; les options de bourrage DOIVENT être ajoutées comme nécessaire pour remplir l'espace d'option jusqu'à une limite de mot. Toutes les options présentes sont incluses dans la somme de contrôle d'en-tête.

Le premier octet d'une option est le type d'option. Les options des types 0 à 31 sont des options d'un seul octet. D'autres options sont suivies par un octet indiquant la longueur de l'option. Cette valeur de longueur inclut les deux octets de type d'option et de longueur d'option ainsi que tous les octets de données d'option ; elle doit donc être supérieure ou égale à 2. Les options DOIVENT être traitées de manière séquentielle, en commençant par la première option dans l'en-tête du paquet. Les options avec des types inconnus DOIVENT être ignorées. En outre, les options avec des longueurs absurdes (longueur de moins de deux octets ou plus que l'espace restant dans la partie options de l'en-tête) DOIVENT être ignorées, et tout l'espace d'option suivant une option avec une longueur insensée doit également être ignoré. Sauf indication contraire, les occurrences multiples de la même option DOIVENT être traitée indépendamment ; pour certaines options, cela signifie en pratique que la dernière occurrence valide d'une option a la priorité.

Les options suivantes sont actuellement définies :

Type	Longueur d'option	Signification	Données DCCP ?	§ de référence
0	1	Bourrage	Oui	5.8.1
1	1	Obligatoire	Non	5.8.2
2	1	Receveur lent	Oui	11.6
3-31	1	Réservé		
32	variable	Changer L	Non	6.1
33	variable	Confirmer L	Non	6.2
34	variable	Changer R	Non	6.1
35	variable	Confirmer R	Non	6.2
36	variable	Init Cookie	Non	8.1.4
37	3-8	Compte NDP	Oui	7.7
38	variable	Vecteur d'ack [Nonce 0]	Non	11.4
39	variable	Vecteur d'ack [Nonce 1]	Non	11.4
40	variable	Données abandonnées	Non	11.7
41	6	Horodatage	Oui	13.1
42	6/8/10	Écho d'horodatage	Oui	13.3
43	4/6	Temps écoulé	Non	13.2
44	6	Données de som. de cont.	Oui	9.3
45-127	variable	Réservé		
128-255	variable	Options spécif. du CCID	-	10.3

Tableau 3 : Options DCCP

Toutes les options ne conviennent pas à tous les types de paquets. Par exemple, comme l'option Vecteur Ack est interprétée par rapport au numéro d'accusé de réception, elle n'est pas appropriée sur les paquets Demande-DCCP et DCCP-Données, qui n'ont pas de numéro d'accusé de réception. Si une option survient sur un type de paquet inattendu, elle DOIT généralement être ignorée ; ces restrictions sont mentionnées dans la description de chaque option. Le tableau résume les restrictions les plus courantes : lorsque la valeur de la colonne DCCP-Données ? est N, l'option correspondante DOIT être ignorée lorsqu'elle est reçue sur un paquet DCCP-Données. (Le paragraphe 7.5.5 explique pourquoi ces options sont ignorées par opposition à, disons, provoquer une réinitialisation.)

Les options avec des valeurs invalides DOIVENT être ignorées, sauf indication contraire. Par exemple, toute option Données de somme de contrôle avec la longueur d'option 4 doit être ignorée, car toutes les options Données de somme de contrôle valables ont une longueur d'option de 6.

Ce paragraphe décrit deux options génériques, Bourrage et Obligatoire. D'autres options sont décrites plus loin.

5.8.1 Option Bourrage

```
+--------+
|00000000|
+--------+
 Type = 0
```

Bourrage est une option d'un seul octet "non-fonctionnement" utilisée pour bourrer entre ou après les options. Si la longueur des autres options d'un paquet n'est pas un multiple de 32 bits, il est alors EXIGÉ des options de bourrage qu'elles remplissent la zone des options jusque à la longueur impliquée par le décalage des données. Le bourrage peut également être utilisé entre les options, par exemple, pour aligner le début d'une option suivante sur une frontière de 32 bits. Il n'y a aucune garantie que les envoyeurs utilisent cette option, de sorte que les receveurs doivent être prêts à traiter les options, même si elles ne commencent pas sur une frontière de mot.

5.8.2 Option Obligatoire

```
+--------+
|00000001|
+--------+
 Type = 1
```

Obligatoire est une option mono-octet qui marque l'option immédiatement suivante comme étant obligatoire. Disons que l'option immédiatement suivante est 0. Par suite, l'option obligatoire n'a aucun effet si le point d'extrémité DCCP receveur comprend et traite 0. Si le point d'extrémité ne comprend ni ne traite 0, il DOIT alors cependant réinitialiser la connexion en utilisant le code Réinitialiser 6, "échec d'obligatoire". Par exemple, le point d'extrémité va réinitialiser la connexion si il ne comprenait pas le type de 0 ; si il a compris le type de 0, mais pas les données de 0 ; si les données de 0 étaient invalides pour le type de 0 ; si 0 était une option de négociation de caractéristique, et si le point d'extrémité n'a pas compris le numéro de caractéristique inclus ; ou si le point d'extrémité a compris 0, mais a choisi de ne pas effectuer l'action que 0 implique. Cette liste n'est pas exhaustive et, en particulier, les spécifications d'options individuelles peuvent décrire des situations supplémentaires dans lesquelles le point d'extrémité devrait réinitialiser la connexion et les situations dans lesquelles il ne le devrait pas.

Les options obligatoire NE DOIVENT PAS être envoyées sur les paquets de données DCCP, et toutes les options Obligatoires reçues sur les paquets de données DCCP DOIVENT être ignorées.

La connexion est en erreur et doit être réinitialisée avec le code Réinitialiser 5,
"Erreur d'option", si l'option 0 est absente (Obligatoire a été le dernier octet de la liste d'options) ou si l'option 0 est égale à Obligatoire. Cependant, la combinaison "Bourrage obligatoire" est valide, et DOIT se comporter comme deux octets de bourrage.
Le paragraphe 6.6.9 décrit le comportement des options de négociation de caractéristique obligatoire plus en détail.

6. Négociation de caractéristique

Quatre options DCCP, Change L, Confirme L, Change R, et Confirme R, sont utilisées pour négocier des valeurs de caractéristiques. Les options Change initient une négociation ; les options Confirme terminent cette négociation. Les options "L" sont envoyées par la localisation de caractéristiques, et les options "R" sont des options envoyées par la caractéristique distante. Les options Change sont retransmises pour assurer la fiabilité.

Toutes ces options ont le même format. Le premier octet des données d'option est le numéro de caractéristique, et le second octet de données et les suivants contiennent une ou plusieurs valeurs de caractéristique. Le format exact de la zone de valeur de caractéristique dépend du type de caractéristique, voir le paragraphe 6.3.

```
+----------+----------+------------------+----------+--------
|   Type   | Longueur |N° caractéristique|Valeur(s)  ...
+----------+----------+------------------+----------+--------
```

Ensemble, le nombre caractéristique et le type d'option ("L" ou "R") identifient de manière unique la caractéristique à laquelle une option s'applique. Le format exact de la zone de la ou des valeurs dépend du numéro de caractéristique.

Les options de négociation de caractéristique NE DOIVENT PAS être envoyées sur les paquets DCCP-Données, et toutes les options de négociation de caractéristique reçues sur des paquets DCCP-Données DOIVENT être ignorées.

6.1 Options Change

Les options Change L et Change R amorcent la négociation de caractéristiques. L'option à utiliser dépend de l'emplacement de la caractéristique concernée : pour démarrer une négociation pour la caractéristique F/A, le DCCP A enverra une option Change L ; pour entamer une négociation pour F/B, il va envoyer une option Change R. Les options Change sont retransmises jusqu'à ce qu'une réponse soit reçue. Elles contiennent au moins une valeur, et ont donc une longueur d'au moins 4.

```
           +--------+--------+--------+--------+--------
Change L: |00100000|  Durée |n° carac| Valeur(s) ...
           +--------+--------+--------+--------+--------
            Type = 32

           +--------+--------+--------+--------+--------
Change R: |00100010| Durée  |n° carac| Valeur(s) ...
           +--------+--------+--------+--------+--------
            Type = 34
```

6.2 Option Confirme

Les options Confirme L et Confirme R terminent la négociation des caractéristiques et sont envoyées en réponse aux options Change R et Change L, respectivement. Les options Confirme NE DOIVENT PAS être générées, sauf en réponse aux options Change. Les options Confirme n'ont pas besoin d'être retransmises, car les options Change sont retransmises si nécessaire. Le premier octet de l'option Confirme contient le numéro de la caractéristique de l'option Change correspondante. Suite à cela est la valeur sélectionnée, puis éventuellement la liste de préférence de l'envoyeur.

```
             +--------+--------+--------+--------+--------
Confirme L : |00100001|Longueur|n° carac| Valeur(s) ...
             +--------+--------+--------+--------+--------
              Type=33

             +--------+--------+--------+--------+--------
Confirme R : |00100011|Longueur|n° carac| Valeur(s) ...
             +--------+--------+--------+--------+--------
              Type=35
```

Si un point d'extrémité reçoit une option Change invalide -- avec un numéro de caractéristique inconnu, ou une valeur non valide -- il répond avec une option Confirme vide contenant le numéro de caractéristique problématique, mais aucune valeur. De telles options ont une longueur de 3.

6.3 Règles de réconciliation

Les règles de réconciliation déterminent comment les deux ensembles de préférences pour une caractéristique donnée sont résolues en un résultat unique. La règle de réconciliation ne dépend que du numéro de caractéristique. Chaque règle de réconciliation doit avoir la propriété que le résultat est déterminé de façon univoque étant donné le contenu des options Change envoyées par les deux points d'extrémité.

Toutes les fonctionnalités actuelles de DCCP utilisent une des deux règles de réconciliation : priorité du serveur ("SP") et non négociable ("NN").

6.3.1 Priorité du serveur

La valeur de caractéristique est une chaîne d'octets de longueur fixe (longueur déterminée par le numéro de caractéristique). Chaque option Change contient une liste de valeurs ordonnée par préférence, la valeur préférée en premier. Chaque option Confirme contient la valeur confirmée, suivie par la liste de préférence de celui qui confirme. Ainsi, la valeur actuelle de la caractéristique va apparaître généralement deux fois dans les données des options Confirme, une fois comme valeur courante et une fois dans la liste de préférence de celui qui confirme.

Pour réconcilier les listes de préférences, sélectionner la première entrée dans la liste du serveur qui apparaît également dans la liste du client. S'il n'y a pas d'entrée partagée, la valeur de la caractéristique NE DOIT PAS changer, et l'option

Confirme confirmera la valeur précédente de la caractéristique (sauf si l'option Change était "Obligatoire" ; voir le paragraphe 6.6.9).

6.3.2 Non négociable

La valeur de caractéristique est une chaîne d'octets. Chaque option contient exactement une valeur de caractéristique. La localisation de caractéristique signale une nouvelle valeur par l'envoi d'une option Change L. La caractéristique distante DOIT accepter toute valeur valide, en répondant avec une option Confirme R contenant la nouvelle valeur, et elle DOIT envoyer des options Confirme R en réponse à des valeurs invalides (sauf si l'option Change L était obligatoire ; voir le paragraphe 6.6.9). Les options Change R et Confirme L NE DOIVENT PAS être envoyées en cas de caractéristiques non négociables ; voir le paragraphe 6.6.8. Les caractéristiques non négociables utilisent le mécanisme de négociation de caractéristique pour réaliser la fiabilité.

6.4 Numéros des caractéristiques

Ce document définit les numéros de caractéristiques suivants.

Numéro	Signification	Règle de réconc.	Valeur initiale	Exigé	§ de référence
0	Réservé				
1	Identifiant de contrôle d'encombrement	SP	2	Oui	10
2	Numéros de séquence courts permis	SP	0	Oui	7.6.1
3	Fenêtre de séquence	NN	100	Oui	7.5.2
4	ECN incapable	SP	0	Non	12.1
5	Taux d'accusés de réception	NN	2	Non	11.3
6	Vecteur d'acc. de réception envoyé	SP	0	Non	11.5
7	Compte de NDP envoyés	SP	0	Non	7.7.2
8	Couverture minimum de s. de contrôle	SP	0	Non	9.2.1
9	Vérifier somme de contrôle de données	SP	0	Non	9.3.1
10-127	Réservé				
128-255	Caractéristiques spécifiques du CCID				10.3

Tableau 4 : Numéros des caractéristiques de DCCP

Règle de réconciliation : règle de la réconciliation utilisée pour la caractéristique. SP signifie serveur prioritaire, NN signifie non négociable.

Valeur initiale : La valeur initiale pour la caractéristique. Chaque caractéristique a une valeur initiale connue.

Exigé : Cette colonne est "Oui" si et seulement si toutes les mises en œuvre de DCCP DOIVENT comprendre la caractéristique. Si c'est "Non", alors la caractéristique se comporte comme une extension (voir Section 15), et il est sûr de répondre aux options Change de la caractéristique avec l'option Confirme vide. Bien sûr, un CCID pourrait exiger la caractéristique ; un DCCP qui met en œuvre CCID 2 DOIT prendre en charge, par exemple, Taux d'accusés de réception et Vecteur d'accusé de réception envoyé.

6.5 Exemples de négociation de caractéristique

Voici trois exemples de négociations de caractéristique pour des caractéristiques situées au serveur, les deux premières pour la caractéristique d'identifiant de contrôle d'encombrement, la dernière pour le taux d'accusés de réception.

Client **Serveur**

1. Change R(CCID, 2 3 1) -->
("2 3 1" est la liste des préférences du client)
2. <-- Confirme L(CCID, 3, 3 2 1)
(3 est la valeur négociée ;
"3 2 1" est la liste des préférences du serveur)

* accord pour que CCID/Serveur = 3 *

1. XXX <-- Change L(CCID, 3 2 1)
2. Retransmission :
<-- Change L(CCID, 3 2 1)
3. Confirme R(CCID, 3, 2 3 1) -->

```
                                * accord pour CCID/Serveur = 3 *

1.                                          <--  Change L(Ack Ratio, 3)
2. Confirme R(Ack Ratio, 3)  -->
                                * accord pour que Ack Ratio/Serveur = 3 *
```

Cet exemple montre une négociation simultanée.

```
     Client                                                    Serveur
1a. Change R(CCID, 2 3 1)  -->
 b.                                         <--  Change L(CCID, 3 2 1)
2a.                                         <--  Confirme L(CCID, 3, 3 2 1)
 b. Confirme R(CCID, 3, 2 3 1)  -->
                                * accord pour que CCID/Serveur = 3 *
```

Voici le codage des octets de plusieurs options Change et Confirme. Chaque option est envoyée par le DCCP A.

Change L (CCID, 2, 3) = 32,5,1,2,3
DCCP B devrait changer la valeur de CCID/A (caractéristique numéro 1, caractéristique de priorité de serveur) ; les valeurs préférées du DCCP A sont 2 et 3, dans cet ordre de préférence.

Change L (fenêtre de séquence, 1024) = 32,9,3,0,0,0,0,4,0
DCCP B devrait changer la valeur de la fenêtre de séquence de A (numéro de caractéristique 3, une caractéristique non négociable) en la chaîne de 6 octets 0,0,0,0,4,0 (la valeur de 1024).

Confirme L (CCID, 2, 2, 3) = 33,6,1,2,2,3
Un DCCP a changé la valeur de CCID/A en 2 ; ses valeurs préférées sont 2 et 3, dans cet ordre de préférence.

Confirme L vide (126) = 33,3,126
DCCP A ne met pas en œuvre le numéro de caractéristique 126, ou la valeur proposée de DCCP B pour la caractéristique 126 / A était invalide.

Change R (CCID, 3 2) = 34,5,1,3,2
DCCP B devrait changer la valeur de CCID/B ; les valeurs préférées de DCCP A sont 3 et 2, dans cet ordre de préférence.

Confirme R (CCID, 2, 3 2) = 35,6,1,2,3,2
Le DCCP a changé la valeur de CCID/B en 2 ; ses valeurs préférées étaient 3 et 2, dans cet ordre de préférence.

Confirme R (fenêtre de séquence, 1024) = 35,9,3,0,0,0,0,4,0
Le DCCP a changé la valeur de la fenêtre de séquence/B en la chaîne de 6 octets 0,0,0,0,4,0 (la valeur 1024).

Confirme R vide (126) = 35,3,126
DCCP A ne met pas en œuvre le numéro de caractéristique 126, ou la valeur proposée du DCCP B pour la caractéristique 126 / B était invalide.

6.6 Échange d'option

Quelques règles de base régissent l'échange d'option de négociation de caractéristique.

1. Chaque option Change non réorganisée obtient une option Confirme en réponse.
2. Les options Change sont retransmises jusqu'à ce qu'une réponse pour le dernier Change soit reçue.
3. Les options de négociation de caractéristique sont traitées dans le strict ordre croissant de numéro de séquence.

Le reste de cette section décrit les conséquences de ces règles plus en détail.

6.6.1 Échange normal

Les options Change sont générées quand un point d'extrémité DCCP veut changer la valeur de certaines caractéristiques. Généralement, cela se fera au début d'une connexion, même si cela peut survenir à tout moment. Nous disons que le point d'extrémité "génère" ou "envoie" une option Change L Change R, mais bien sûr l'option doit être jointe à un paquet. Le point d'extrémité peut joindre l'option à un paquet qu'il aurait généré toute façon (comme une Demande-DCCP) ou il peut créer un "paquet de négociation de caractéristique", souvent un DCCP-Ack ou un DCCP-Sync, juste pour porter l'option. Les paquets de négociation de caractéristiques sont contrôlés par le mécanisme de contrôle d'encombrement pertinent. Par

exemple, le DCCP A ne peut envoyer un DCCP-Ack ou DCCP-Sync pour la négociation de caractéristique que si le CCID B à A permettrait l'envoi d'un DCCP-Ack. En outre, un point d'extrémité doit générer au plus un paquet de négociation de caractéristiques par temps d'aller-retour.

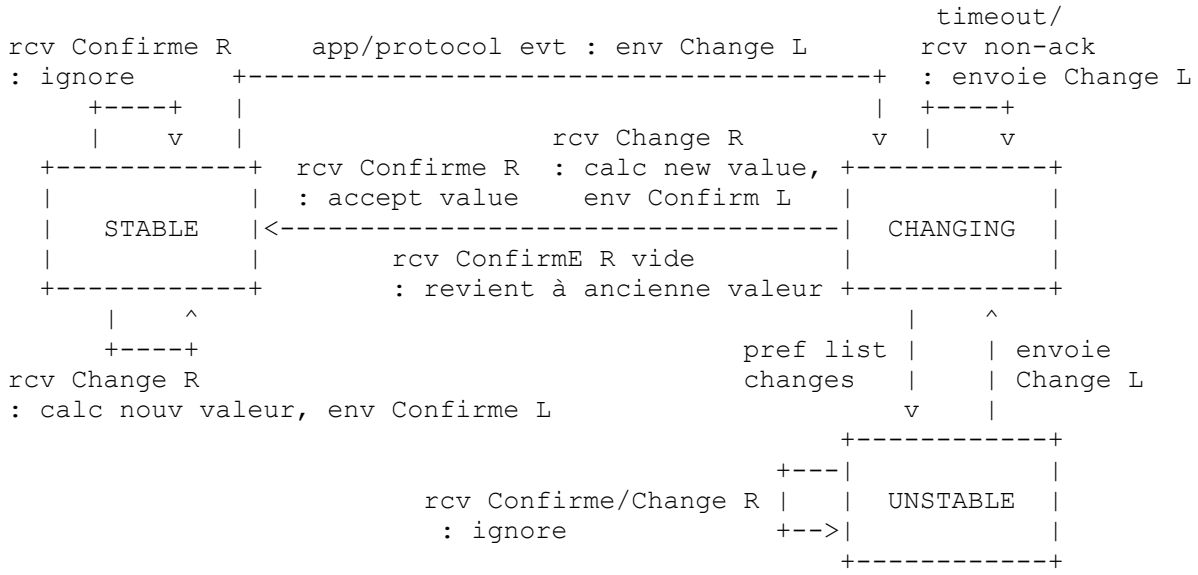
À réception d'une option Change L ou Change R, un point d'extrémité DCCP examine la liste des préférences incluse, la concilie avec sa propre liste de préférences, calcule la nouvelle valeur, et envoie en retour une option Confirme R ou Confirme L, respectivement, informant ses homologues de la nouvelle valeur ou que la caractéristique n'a pas été comprise. Chaque option Change non réordonnée DOIT aboutir à une option Confirme correspondante, et tout paquet incluant une option Confirme doit porter un numéro d'accusé de réception. (Le paragraphe 6.6.4 décrit comment est détecté et traité un Change réordonné.) Les options Confirme générées peuvent être attachées aux paquets qui auraient été tout de même envoyés (comme les DCCP-Réponse ou les DCCP-SyncAck) ou à de nouveaux paquets de négociation de caractéristiques, comme décrit ci-dessus.

Le point d'extrémité qui envoie le Change DOIT attendre de recevoir une option Confirme correspondante avant de changer sa valeur de caractéristique mémorisée. Le point d'extrémité qui envoie le Confirme change sa valeur de caractéristique mémorisée dès qu'il envoie le Confirme.
Un paquet PEUT contenir plus d'une option de négociation de caractéristiques, éventuellement en incluant deux options qui se réfèrent à la même caractéristique ; comme d'habitude, les options sont traitées de manière séquentielle.

6.6.2 Traitement des options reçues

Les points d'extrémité DCCP existent dans l'un des trois états relatifs à chaque caractéristique. STABLE est l'état normal, où le point d'extrémité connaît la valeur de la caractéristique et pense que l'autre extrémité est d'accord. Un point d'extrémité entre dans l'état CHANGING quand il envoie d'abord un Change pour la caractéristique et retourne à STABLE une fois qu'il reçoit un Confirme correspondant. L'état final, UNSTABLE, indique qu'un point d'extrémité dans l'état CHANGING a changé sa liste de préférences, mais n'a pas encore transmis une option Change avec la nouvelle liste de préférences.

Les transitions d'état de caractéristiques à une localisation de caractéristiques sont mises en œuvre selon le schéma ci-dessous. Le schéma ignore le numéro de séquence et les questions de validité d'option ; elles sont traitées explicitement dans le pseudo-code qui suit.



Les localisations de caractéristique DEVRAIENT utiliser le pseudo-code suivant, qui correspond au diagramme d'état, pour réagir à chaque option de négociation de caractéristique sur chaque paquet valide reçu qui n'est pas de données. Le pseudocode se réfère à "P.seqno" et "P.ackno", qui sont des propriétés du paquet, à "0.type" et "0.len", qui sont des propriétés de l'option, à "FGSR" et "FGSS", qui sont les propriétés de la connexion et traitent la réorganisation comme décrit au paragraphe 6.6.4, à "F.state", qui est l'état de la caractéristique (STABLE, CHANGING ou UNSTABLE), et à "F.value", qui est la valeur de la caractéristique.

Tout d'abord, vérifier les caractéristiques inconnues (paragraphe 6.6.7) ;
Si F est inconnue,
Si l'option était obligatoire, / * paragraphe 6.6.9 * /
Réinitialisation de la connexion et retour

```
      Sinon, si 0.type  == Change R,
        Envoyer Confirmer L vide  sur un paquet à venir

      Retour

Deuxièmement, vérifier la réorganisation (paragraphe 6.6.4) ;
    Si F.state == instable ou P.seqno ? FGSR
         ou (0.type == Confirme R et P.ackno < FGSS),
      Ignorer l'option et retour

Troisièmement, traiter les options Change R ;
    Si 0.type == Change R,
      Si la valeur de l'option est valide,   / * paragraphe 6.6.8 * /
        Calculer la nouvelle valeur
        Envoyer Confirme L sur un paquet à venir
        Régler F.state: = stable
      Sinon, si l'option était obligatoire,
        Réinitialisation de la connexion et le retour
      Sinon,
        Envoyer Confirme L vide sur un paquet à venir
/ * Restez dans l'état actuel. Si c'est un CHANGING, ce point d'extrémité va retransmettre son option Change L plus tard. *
/

Quatrièmement, traiter les options Confirme R (mais seulement dans l'état CHANGING).
    Si F.state == CHANGING et 0.type == Confirme R,
      Si 0.len > 3, / * non vide * /
        Si la valeur de l'option est valide,
          Régler F.value: = nouvelle valeur
        Sinon,
          Réinitialisation de la connexion et retour
      Régler  F.state: = STABLE
```

Des versions de ce schéma et du pseudo code sont également utilisées par les caractéristique distantes ; il suffit d'échanger les "L" et les "R", de sorte que les options pertinentes soient Change R et Confirme L.

6.6.3 Perte et retransmission

Les paquets contenant les options Change et Confirme pourraient être perdus ou retardés par le réseau. Par conséquent, les options Change sont transmises de façon répétée pour réaliser la fiabilité. On se réfère à cela sous le nom de "retransmission", même si bien sûr il n'y a pas de retransmission au niveau des paquets dans DCCP : une option Change qui est envoyée à nouveau sera envoyée sur un nouveau paquet avec un nouveau numéro de séquence.

Un point d'extrémité CHANGING transmet une autre option Change une fois qu'il se rend compte qu'il n'a pas eu de réponse de l'autre extrémité. La nouvelle option Change ne doit pas contenir la même charge utile que l'originale ; la protection de réorganisation veillera à ce que l'accord se fasse sur la base de l'option la plus récemment transmise.

Un point d'extrémité CHANGING DOIT continuer a retransmettre les options Change jusqu'à ce il obtienne une réponse ou que la connexion se termine.

Les points d'extrémité DEVRAIENT utiliser un temporisateur à retard exponentiel pour décider quand retransmettre les options Change. (Les paquets générés spécifiquement pour les négociations de caractéristiques DOIVENT utiliser un tel temporisateur.) L'intervalle de temporisation est initialement fixé à au moins un délai d'aller-retour, et devrait revenir à pas moins de 64 secondes. Le retard protège contre un accord différé en raison des algorithmes de protection contre la réorganisation décrits dans le paragraphe suivant. Encore une fois, les terminaux peuvent porter les options Change sur des paquets qu'ils auraient tout de même envoyés, ou créer de nouveaux paquets pour porter les options. Tous les nouveaux paquets sont contrôlés par le mécanisme de contrôle d'encombrement pertinent.

Les options Confirme ne sont jamais retransmises, mais le point d'extrémité qui envoie le Confirme DOIT générer une option Confirme après chaque Change non réordonné.

6.6.4 Réorganisation

Une réorganisation pourraient être cause que des paquets contenant les options Change et Confirme arrivent dans un ordre inattendu. Les points d'extrémité DOIVENT ignorer les options de négociation de caractéristique qui ne sont pas arrivés en ordre strictement croissant de numéro de séquence. Le reste de cette section présente deux algorithmes qui répondent à cette exigence.

Le premier algorithme introduit deux variables de numéro de séquence que chaque extrémité entretient pour la connexion.

FGSR *(Feature Greatest Sequence Number Received)* plus grand numéro de séquence de caractéristique reçu :
Plus grand numéro de séquence reçu, en considérant seulement des paquets valides qui contenaient une ou plusieurs options de négociation de caractéristique (Change et/ou Confirme). Cette valeur est initialisée à ISR - 1.

FGSS *(Feature Greatest Sequence Number Sent)* plus grand numéro de séquence de caractéristique envoyé :
Plus grand numéro de séquence envoyé, en considérant uniquement les paquets qui contenaient une ou plusieurs options Change nouvelles. Une option Change est nouvelle si et seulement si elle a été produite pendant une transition de l'état STABLE ou UNSTABLE à l'état CHANGING ; les options Change générées au sein de l'état CHANGING sont des retransmissions et DOIVENT avoir exactement le même contenu que les options précédemment transmises, permettant une tolérance de réorganisation. FGSS est initialisé à ISS.

Chaque point d'extrémité vérifie les deux conditions sur les numéros de séquence pour décider s'il faut traiter les options de négociation de caractéristique reçues.

1. Si le numéro de séquence d'un paquet est inférieur ou égal à FGSR, ses options Change DOIVENT être ignorées.

2. Si le numéro de séquence d'un paquet est inférieur ou égal à FGSR, si il n'a pas de numéro d'accusé de réception, OU si son numéro d'accusé de réception est inférieur au FGSS, alors ses options Confirme DOIVENT être ignorées.

Autrement, un point d'extrémité PEUT tenir des valeurs FGSR et FGSS séparées pour chaque caractéristique. FGSR (F/X) serait égal au plus grand numéro de séquence reçu, en considérant uniquement les paquets qui contenaient des options Change ou Confirme s'appliquant à la caractéristique F/X ; FGSS (F/X) serait défini de façon similaire. Cet algorithme nécessite plus d'état, mais est légèrement plus permissif pour des négociations de caractéristique multiples qui se chevauchent. L'un ou l'autre algorithme PEUT être utilisé ; le premier algorithme, avec des variables FGSR et FGSS à l'échelle de la connexion, est RECOMMANDÉ.

Une conséquence de ces règles est qu'un point d'extrémité dans l'état CHANGING ignorera toute option Confirme qui n'accuse pas réception de la dernière option Change envoyée. Cela garantit que l'accord, une fois réalisé, a utilisé les plus récentes informations disponibles au sujet des préférences des points d'extrémité.

6.6.5 Changements de préférences

Les points d'extrémité sont autorisés à modifier leurs listes de préférences à tout moment. Cependant, un point d'extrémité qui change sa liste de préférence quand il est dans l'état CHANGING DOIT passer à l'état UNSTABLE. Il va revenir à l'état CHANGING une fois qu'il aura transmis une option Change avec la nouvelle liste de préférences. Cela garantit que l'accord est fondé sur les listes de préférence actives. Sans l'état UNSTABLE, la négociation simultanée -- où les points d'extrémité ont entamé des négociations indépendantes pour la même caractéristique au même moment -- pourrait conduire à la terminaison de la négociation, les extrémités pensant que la caractéristique avait une valeur différente.

6.6.6 Négociation simultanée

Les deux extrémités peuvent ouvrir simultanément des négociations pour la même caractéristique, après quoi un point d'extrémité dans l'état CHANGING va recevoir une option Change pour la même caractéristique. De telles options Change reçues peuvent agir comme réponses aux options Change d'origine. Le point d'extrémité dans l'état CHANGING DOIT examiner la liste de préférences de Change reçue, les concilier avec sa propre liste de préférences (telle qu'exprimée dans son option Change générée) et générer l'option Confirme correspondante. Il peut ensuite effectuer la transition à l'état STABLE.

6.6.7 Caractéristiques inconnues

Les points d'extrémité peuvent recevoir des options Change se référant à des numéros de caractéristique qu'ils ne comprennent pas -- par exemple, quand un DCCP étendu converse avec un DCCP non étendu. Les points d'extrémité DOIVENT répondre aux options Change inconnues avec des options Confirme vides (c'est-à-dire des options Confirme

vides ne contenant pas de données) qui informent le point d'extrémité CHANGING que la caractéristique n'a pas été comprise. Toutefois, si l'option Change était Obligatoire, la connexion DOIT être réinitialisé, voir au paragraphe 6.6.9.

À réception d'une option Confirme vide pour une caractéristique, le point d'extrémité CHANGING DOIT repasser à l'état STABLE, laissant la valeur de la caractéristique inchangée. La Section 15 indique que la valeur par défaut pour toute caractéristique d'extension correspond à "extension non disponible".

Il est exigé de certaines caractéristiques qu'elles soient comprises par tous les DCCP (voir le paragraphe 6.4). Le point d'extrémité CHANGING DEVRAIT réinitialiser la connexion (avec le code de réinitialisation 5, "Erreur d'option") si il reçoit une option Confirme vide pour une telle caractéristique.

Comme les options Confirme sont générées uniquement en réponse aux options Change, un point d'extrémité ne devrait jamais recevoir une option Confirme se référant à un numéro de caractéristique qu'il ne comprend pas. Néanmoins, les points d'extrémité DOIVENT ignorer de telles options si ils en reçoivent.

6.6.8 Options invalides

Un point d'extrémité DCCP pourrait recevoir une option Change ou Confirme pour une caractéristique connue qui contient une ou plusieurs valeurs qu'il ne comprend pas. Certaines, mais pas toutes, de ces options ne sont pas valides, selon la règle de réconciliation pertinente (paragraphe 6.3). Par exemple :

- Toutes les caractéristiques ont des limitations de longueur, et les options avec des longueurs non valides sont invalides. Par exemple, la caractéristique Ratio Ack prend des valeurs de 16 bits, donc les options "Confirme R (ratio Ack)" valables ont une longueur d'option de 5.

- Certaines caractéristiques non négociables ont des limites de valeur. La caractéristique Ack Ratio prend des valeurs d'entier de deux octets, non nulles, donc une option "Change L (Ratio Ack, 0)" n'est jamais valable. Notez que les caractéristiques de priorité de serveur n'ont pas de limitation de valeur, puisque les valeurs inconnues sont traitées normalement.

- Toute option Confirme qui choisit la mauvaise valeur, sur la base des deux listes de préférences et de la règle de réconciliation pertinente, n'est pas valide.

Toutefois, les options Confirme inattendues -- qui se réfèrent à des numéros de caractéristique inconnus, ou qui ne semblent pas faire partie d'une négociation en cours -- ne sont pas invalides, mais elles sont ignorées par le receveur.

Un point d'extrémité qui reçoit une option Change invalide DOIT répondre avec l'option Confirme correspondante vide. Un point d'extrémité recevant une option Confirme invalide DOIT réinitialiser la connexion, avec le code de réinitialisation 5, "Erreur d'option".

6.6.9 Négociation de caractéristique obligatoire

Les options Change peuvent être précédées par des options obligatoires (paragraphe 5.8.2).
Les options Change obligatoires sont traitées comme des options de changement normales sauf que les cas d'échec suivants vont faire que le receveur va réinitialiser la connexion avec le code de réinitialisation 6, "Échec obligatoire", plutôt que d'envoyer une option Confirme. La connexion DOIT être remise à zéro si :
- le numéro de caractéristique de l'option Change n'a pas été compris ;
- la valeur de l'option Change était invalide, et le receveur aurait normalement envoyé une option Confirme vide en réponse, ou
- pour les caractéristiques de priorité de serveur, il n'y avait pas d'entrée commune aux deux listes de préférences des points d'extrémité.

Les autres cas d'échec ne causent pas de réinitialisation de la connexion, en particulier, la protection contre la réorganisation peut être cause qu'une option Change obligatoire sera ignorée sans réinitialiser la connexion.

Les options Confirme se comportent de manière identique et ont les mêmes conditions de réinitialisation qu'elles soient ou non obligatoires.

7. Numéros de séquence

DCCP utilise des numéros de séquence pour ranger les paquets en séquence, pour détecter les pertes et les doublons du réseau, et pour protéger contre les attaquants, les connexions semi-ouvertes, et la livraison de très vieux paquets. Chaque paquet porte un numéro de séquence ; la plupart des types de paquets portent aussi un numéro d'accusé de réception.

Les numéros de séquence DCCP sont fondés sur le paquet. C'est-à-dire que les numéros de séquence générés par chaque point d'extrémité augmentent de un, modulo 2^48, par paquet. Même les paquets DCCP-Ack et DCCP-Sync, et d'autres qui ne transportent pas de données d'utilisateur, incrémentent le numéro de séquence. Comme DCCP est un protocole non fiable, il n'y a pas de vraies retransmissions, mais les retransmissions effectives, telles que les retransmissions des paquets Demande-DCCP, incrémentent également le numéro de séquence. Cela permet aux mises en œuvre DCCP de détecter la duplication sur le réseau, les retransmissions, et les pertes d'accusé de réception ; c'est une différence significative avec la pratique de TCP.

7.1 Variables

Les points d'extrémité DCCP tiennent un ensemble de variables de numéro de séquence pour chaque connexion.

ISS Le numéro de séquence initial envoyé par ce point d'extrémité. Il est égal au numéro de séquence de la première Demande-DCCP ou DCCP-Réponse envoyée.

ISR Le numéro de séquence initial reçu de l'autre point d'extrémité. Cela équivaut au numéro de séquence de la première Demande-DCCP ou DCCP-Réponse reçue.

GSS Le plus grand numéro de séquence envoyé par ce point d'extrémité. Ici, et d'ailleurs, "le plus grand" est mesuré dans un espace de séquence circulaire.

GSR Le plus grand numéro de séquence reçu de l'autre point d'extrémité sur un paquet acquittable. (Ce terme est défini au paragraphe 7.4).

GAR Le plus grand numéro d'accusé de réception reçu de l'autre point d'extrémité sur un paquet acquittable qui n'était pas un DCCP-Sync.

D'autres variables sont dérivées de ces primitives.

SWL et SWH : (Fenêtre basse/haute de numéro de séquence.) Les extrêmes de la fenêtre de validité des numéros de séquence des paquets reçus.

AWL et AWH : (Fenêtre basse/haute de numéro d'accusé de réception) Les extrêmes de la fenêtre de validité pour les numéros d'accusé de réception des paquets reçus.

7.2 Numéros de séquence initiaux

Les numéros de séquence initiaux des points d'extrémité sont fixés par les premiers paquets Demande-DCCP et DCCP-Réponse envoyés. Les numéros de séquence initiaux DOIVENT être choisis pour éviter deux problèmes :

- o la livraison de paquets anciens, où les paquets qui subsistent dans le réseau à partir d'une ancienne connexion sont livrés à une nouvelle connexion avec les mêmes adresses et numéros d'accès,
- o les attaques de numéro de séquence, où un attaquant peut deviner les numéros de séquence qu'une connexion future va utiliser [M85].

Ces problèmes sont les mêmes que ceux rencontrés par TCP, et les mises en œuvre de DCCP DEVRAIENT utiliser les stratégies de TCP pour les éviter [RFC0793], [RFC1948]. Le reste de cette section explique ces stratégies plus en détail.

Pour résoudre le premier problème, une mise en œuvre DOIT s'assurer que le numéro de séquence initial pour un quadruplet donné <adresse de source, accès de source, adresse de destination, accès de destination> ne se chevauche pas avec les numéros de séquence récents sur des connexions précédentes avec le même quadruplet. ("récents" signifie envoyés dans deux durées de vie du segment au maximum, soit 4 minutes.) La mise en œuvre DOIT de plus s'assurer que les 24 bits de moindre poids du numéro de séquence initial ne se chevauchent pas avec les 24 bits de moindre poids des numéros de séquence récents (à moins que la mise en œuvre prévoit d'éviter les numéros de séquence courts, voir au paragraphe 7.6). Une mise en œuvre qui a l'état d'une connexion récente avec le même quadruplet peut choisir explicitement un bon numéro de séquence initial. Sinon, elle pourrait lier le choix du numéro de séquence initial à une horloge, comme l'horloge de 4

microsecondes utilisée par TCP [RFC0793]. Deux horloges distinctes peuvent être nécessaires, une pour les 24 bits de poids fort et une pour les 24 bits de moindre poids.

Pour répondre au second problème, une mise en œuvre DOIT fournir à chaque quadruplet un espace indépendant de numéro de séquence initial. Ensuite, l'ouverture d'une connexion ne fournit aucune information sur les numéros de séquence initiaux sur les autres connexions au même hôte. La [RFC1948] y parvient en ajoutant un hachage cryptographique du quadruplet et un secret pour chaque numéro de séquence initial. Pour le secret, la [RFC1948] recommande une combinaison de certaines données véritablement aléatoires [RFC4086], un mot de passe installé administrativement, l'adresse IP du point d'extrémité, et l'heure du démarrage du point d'extrémité, mais des données véritablement aléatoires sont suffisantes. On devrait faire attention lorsque le secret est changé ; un tel changement modifie tous les espaces de numéro de séquence initial, ce qui pourrait rendre un numéro de séquence initial pour certains quadruplets égal à un numéro de séquence envoyé récemment pour le même quadruplet. Pour éviter ce problème, le point d'extrémité peut se souvenir de l'état de la connexion morte pour chaque quadruplet ou rester au repos pour au maximum deux durées de vie de segment autour d'un tel changement.

7.3 Temps de repos

Les points d'extrémité DCCP, comme les points d'extrémité TCP, doivent faire attention avant d'initier les connexions lors de l'amorçage. En particulier, ils NE DOIVENT PAS envoyer des paquets dont les numéros de séquence sont proches des numéros de séquence de paquets qui subsistent dans le réseau depuis avant l'amorçage. Le plus simple moyen de faire appliquer cette règle est que les points d'extrémité DCCP évitent d'envoyer tout paquet pendant une durée de vie maximale du segment (2 minutes) après l'amorçage.

D'autres mécanismes d'application comprennent de se souvenir des numéros de séquence récents d'un amorçage à l'autre et de réserver les 8 bits de poids fort des numéros de séquence initiaux pour un compteur persistant qui décrémente de deux à chaque amorçage. (Le dernier mécanisme exigerait d'interdire les paquets avec des numéros de séquence courts ; voir au paragraphe 7.6.1).

7.4 Numéros d'accusé de réception

Les accusés de réception cumulatifs n'ont pas de sens dans un protocole non fiable. Par conséquent, le champ DCCP Numéro d'accusé de réception a un sens différent de celui de TCP.

Un paquet reçu est classé comme acquittable si et seulement si son en-tête a été traité avec succès par le DCCP receveur. Dans les termes du pseudo code du paragraphe 8.5, un paquet reçu devient acquittable lorsque le point d'extrémité receveur parvient à l'étape 8. Cela signifie, par exemple, que tous les paquets acquittables ont des sommes de contrôle d'en-tête et des numéros de séquence valides. Le numéro d'accusé de réception d'un paquet envoyé DOIT être égal au GSR du point d'extrémité envoyeur, le plus grand numéro de séquence reçu sur un paquet acquittable, pour tous les types de paquets, sauf DCCP-Sync et DCCP-SyncAck.

"Acquittable" ne se réfère pas au traitement des données. Même les paquets acquittables peuvent voir leurs données d'application éliminées, en raison du débordement de la mémoire tampon de réception ou de corruption, par exemple. Les rapports d'option de données éliminées rapportent ces pertes de données lorsque cela est nécessaire, laissant les mécanisme de contrôle d'encombrement faire la distinction entre les pertes du réseau et les pertes de point d'extrémité. Cette question est examinée plus en détails aux paragraphes 11.4 et 11.7.

Les numéros d'accusé de réception des paquets DCCP-Sync et DCCP-SyncAck diffèrent comme suit : le numéro d'accusé de réception sur un paquet DCCP-Sync correspond à un paquet reçu, mais pas nécessairement à un paquet acquittable ; en particulier, il pourrait correspondre à un paquet désynchronisé dont les options n'ont pas été traitées. Le numéro d'accusé de réception sur un paquet DCCP-SyncAck correspond toujours à un paquet DCCP-Sync acquittable ; il pourrait être inférieur à GSR, en présence de réorganisation.

7.5 Validité et synchronisation

Tout point d'extrémité DCCP pourrait recevoir des paquets qui ne font pas réellement partie de la connexion en cours. Par exemple, le réseau pourrait livrer un ancien paquet, un attaquant pourrait tenter de capturer une connexion, ou l'autre point d'extrémité peut avoir une défaillance, provoquant une connexion semi-ouverte.

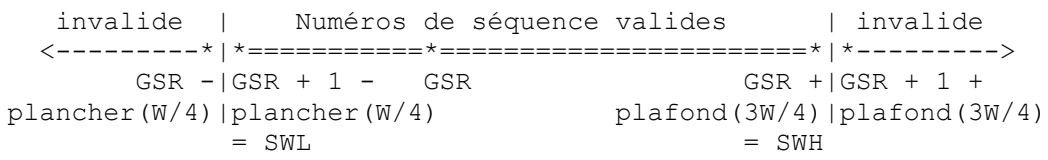
DCCP, comme TCP, utilise les vérifications de numéro de séquence pour détecter ces cas. Les paquets dont le numéro de séquence et/ou d'accusé de réception est hors gamme sont appelées des séquences non valides et ne sont pas traités normalement.

Contrairement à TCP, DCCP nécessite un mécanisme de synchronisation pour se remettre de grandes salves de pertes. Un point d'extrémité pourrait envoyer tant de paquets au cours d'une salve de pertes que lorsque l'un de ces paquets passe finalement au travers, l'autre point d'extrémité va marquer son numéro de séquence comme invalide. Une prise de contact de paquets DCCP-Sync et DCCP-SyncAck récupère de ces cas.

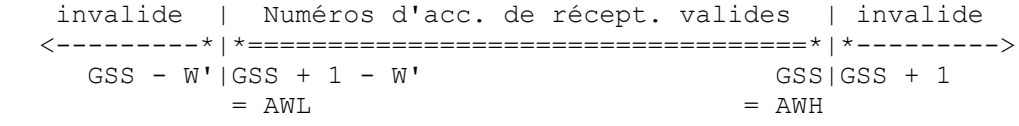
7.5.1 Fenêtre de numéro de séquence et d'accusé de réception

Chaque point d'extrémité DCCP définit des fenêtres de validité de séquence qui sont des sous-ensembles des espaces de numéros de séquence et d'accusé de réception. Ces fenêtres correspondent à des paquets que le point d'extrémité s'attend à recevoir dans les prochains temps d'aller-retour. Les fenêtres de numéro de séquence et d'accusé de réception contiennent toujours, respectivement, le GSR et le GSS. Les largeurs de fenêtre sont contrôlées par des caractéristiques de fenêtre de séquence pour les deux demi-connexions.

La fenêtre de validité de numéro de séquence pour les paquets à partir de DCCP B est [SWL, SWH]. Cette fenêtre contient toujours le GSR, le plus grand numéro de séquences reçu sur un paquet à séquence valide de DCCP B. Il est large de W paquets, où W est la valeur de la caractéristique Fenêtre de séquence/B. Un quart de la fenêtre de séquence, arrondi à l'entier inférieur, est inférieur ou égal au GSR, et les trois-quarts sont plus grands que le GSR. (Cette répartition asymétrique suppose que des rafales de perte sont plus fréquentes dans le réseau que les réorganisations significatives.)



La fenêtre de la validité des numéros d'accusé de réception des paquets provenant de DCCP B est [AWL, AWH]. Le sommet de la fenêtre, AWH, équivaut au GSS, le plus grand numéro de séquence envoyé par DCCP A ; la fenêtre est large de W' paquets, où W' est la valeur de la caractéristique Fenêtre de séquence /A.



SWL et AWL sont initialement ajustés de telle sorte qu'ils ne soient pas inférieurs aux numéros de séquence initiaux, respectivement reçus et envoyés :

SWL = max(GSR + 1 - plancher(W/4), ISR),
AWL = max(GSS + 1 - W', ISS).

Ces ajustements NE DOIVENT être appliqués qu'au début de la connexion. (Des connexions à longue durée de vie peuvent faire revenir à zéro les numéros de séquence de sorte qu'ils semblent être inférieurs à l'ISR ou à l'ISS ; les ajustements NE DOIVENT PAS être appliqués dans ce cas.)

7.5.2 Caractéristique de fenêtre de séquence

La caractéristique Fenêtre de séquence/A détermine la largeur de la fenêtre de validité de numéro de séquence utilisée par DCCP B et la largeur de la fenêtre de validité des accusés de réception utilisée par DCCP A. Le DCCP A envoie une option "Change L(fenêtre de séquence, W)" pour notifier au DCCP B que la valeur de la fenêtre de séquence/A est W.

La fenêtre de séquence a le numéro de caractéristique 3 et est non négociable. Elle prend des valeurs d'entiers de 48 bits (6 octets) comme les numéros de séquence DCCP. Les options Change et Confirme pour les fenêtres de séquence sont donc de 9 octets de long. Les nouvelles connexions commencent avec la fenêtre de séquence 100 pour les deux points d'extrémité. La valeur minimale de fenêtre de séquence valide est Wmin = 32. La valeur maximum de fenêtre de séquence est Wmax = 2 ^ 46 - 1 = 70 368 744 177 663. Les options Change suggérant des valeurs de fenêtre de séquence hors de cette plage sont invalides et DOIVENT être traitées en conséquence.

Une valeur de fenêtre de séquence/A appropriée doit refléter le nombre de paquets que DCCP A s'attend à être en cours. Seul DCCP A peut anticiper ce nombre. Les valeurs qui sont trop petites augmentent le risque que les points d'extrémité perdent la synchronisation après des salves de pertes, et les valeurs qui sont beaucoup trop petites peuvent empêcher une communication productive qu'il y ait ou non des pertes. D'autre part, les valeurs trop grandes augmentent le risque de

capture de connexion ; le paragraphe 7.5.5 quantifie ce risque. Une bonne ligne directrice est que pour chaque point d'extrémité on règle la fenêtre de séquence à environ cinq fois le nombre maximal de paquets qu'on s'attend à envoyer dans un délai d'aller retour. Les points d'extrémité DEVRAIENT envoyer des options Change L (fenêtre de séquence) comme nécessaire, lorsque la connexion progresse. En outre, un point d'extrémité NE DOIT PAS systématiquement envoyer plus que son nombre de fenêtre de séquence de paquets par temps d'aller-retour ; c'est-à-dire que DCCP A NE DOIT PAS envoyer de façon persistante plus de paquets que la fenêtre de séquence/A par RTT.

7.5.3 Règles de validité de séquence

La validité de séquence dépend du type du paquet reçu. Ce tableau montre les vérifications de numéro de séquence et d'accusé de réception appliqués à chaque paquet ; un paquet n'a une séquence valide que si il réussit les deux essais, et a une séquence invalide si il n'y réussit pas. Beaucoup de contrôles se réfèrent à la fenêtre de validité du numéro de séquence et d'accusé de réception [SWL, SWH] et [AWL, AWH] définis au paragraphe 7.5.1.

Type de paquet	Vérification du numéro de séquence	Vérification du numéro de séquence
Demande-DCCP	SWL ? seqno ? SWH (*)	N/A
DCCP-Réponse	SWL ? seqno ? SWH (*)	AWL ? ackno ? AWH
DCCP-Données	SWL ? seqno ? SWH	N/A
DCCP-Ack	SWL ? seqno ? SWH	AWL ? ackno ? AWH
DCCP-DataAck	SWL ? seqno ? SWH	AWL ? ackno ? AWH
DCCP-CloseReq	GSR < seqno ? SWH	GAR ? ackno ? AWH
DCCP-Close	GSR < seqno ? SWH	GAR ? ackno ? AWH
DCCP-Reset	GSR < seqno ? SWH	GAR ? ackno ? AWH
DCCP-Sync	SWL ? seqno	AWL ? ackno ? AWH
DCCP-SyncAck	SWL ? seqno	AWL ? ackno ? AWH

(*) Vérification non appliquée si la connexion est en état LISTEN ou REQUEST.

En général, les paquets sont valables en séquence si leurs numéros de séquence et d'accusés de réception se situent dans les fenêtres valides correspondantes, [SWL, SWH] et [AWL, AWH]. Les exceptions à cette règle sont les suivantes :

- Comme les paquets DCCP-CloseReq, DCCP-Close, et DCCP-Reset terminent une connexion, ils ne peuvent pas avoir des numéros de séquence inférieurs ou égaux au GSR, ou des numéros d'accusé de réception inférieurs au GAR.
- Les numéros de séquence DCCP-Sync et DCCP-SyncAck ne sont pas fortement vérifiés. Ces types de paquets existent spécifiquement pour remettre les points d'extrémité en synchronisation ; vérifier leurs numéros de séquence éliminerait leur utilité.

Les faibles contrôles sur les paquets DCCP-Sync et DCCP-SyncAck paquets permettent le maintien du fonctionnement après des événements inhabituels, comme les défaillances de point d'extrémité et les grandes salves de pertes, mais il n'y a pas besoin d'indulgence en l'absence d'événement extraordinaire -- c'est-à-dire, pendant une communication réussie en cours. Par conséquent, les mises en œuvre DCCP DEVRAIENT utiliser les vérifications suivantes, plus rigoureuses pour les connexions actives, où une connexion est considérée comme active si elle a reçu des paquets valides à partir l'autre point d'extrémité dans les trois derniers temps d'aller-retour.

Type de paquet	Vérification du numéro de séquence	Vérification du numéro de séquence
DCCP-Sync	SWL ? seqno ? SWH	AWL ? ackno ? AWH
DCCP-SyncAck	SWL ? seqno ? SWH	AWL ? ackno ? AWH

Enfin, un point d'extrémité peut appliquer les contrôles plus stricts pour la suite DCCP-CloseReq, DCCP-Close, et DCCP-Reset paquets, abaissant encore la probabilité de succès des attaques aveugles qui utilisent ces types de paquets.

Étant donné que ces contrôles peuvent provoquer des frais généraux de synchronisation supplémentaires et retarder la fermeture de la connexion lorsque les paquets sont perdus, ils devraient être considérés comme expérimentaux.

Type de paquet	Vérification du numéro de séquence	Vérification du numéro d'accusé de réception
DCCP-CloseReq	seqno == GSR + 1	GAR ? ackno ? AWH
DCCP-Close	seqno == GSR + 1	GAR ? ackno ? AWH
DCCP-Reset	seqno == GSR + 1	GAR ? ackno ? AWH

Noter que la validité de séquence est seulement l'un des contrôles de validité appliqués aux paquets reçus.

7.5.4 Traitement des paquets au numéro de séquences invalide

Les points d'extrémité répondent aux paquets reçus de séquence invalide comme suit.

- o Tout paquet DCCP-Sync ou DCCP-SyncAck de séquence invalide DOIT être ignoré.
- o Un paquet DCCP-Reset de séquence invalide DOIT choisir un paquet DCCP-Sync en réponse (sous réserve d'une possible limite de débit). Ce paquet de réponse DOIT utiliser un nouveau numéro de séquence, et va donc augmenter le GSS ; le GSR ne changera cependant pas, puisque le paquet reçu a un numéro de séquence invalide. Le numéro d'accusé de réception du paquet de réponse DOIT être égal au GSR.
- o Tout autre paquet de numéro de séquence invalide DOIT choisir un paquet DCCP-Sync similaire, sauf que le numéro d'accusé de réception du paquet de réponse DOIT être égal au numéro de séquence du paquet invalide.

À réception d'un paquet DCCP-Sync valide en séquence, le point d'extrémité homologue (disons, DCCP B) DOIT mettre à jour sa variable GSR et répondre avec un paquet DCCP-SyncAck. Le numéro d'accusé de réception du paquet DCCP-SyncAck sera égal au numéro de séquence du DCCP-Sync, qui n'est pas nécessairement le GSR. Dès réception de ce DCCP-SyncAck, qui sera valable en séquence car il accuse réception du DCCP-Sync, le DCCP A mettra à jour sa variable GSR, et les points d'extrémités seront de retour en synchronisation. Par exception, si le point d'extrémité homologue est dans l'état Demande, il DOIT répondre avec un DCCP-Reset au lieu d'un DCCP-SyncAck. Cela sert à nettoyer la connexion demi-ouverte du DCCP A.

Pour se protéger contre les attaques de déni de service, les mises en œuvre DCCP DEVRAIENT imposer une limite de débit aux DCCP-Sync envoyés en réponse aux paquets à séquence invalide, telle que pas plus de huit DCCP-Sync par seconde.

Les points d'extrémité DCCP NE DOIVENT PAS traiter les paquets de séquence invalide, sauf peut-être, en générant un DCCP-Sync. Par exemple, les options NE DOIVENT PAS être traitées. Un point d'extrémité PEUT temporairement conserver des paquets à séquence invalide pour le cas où ils deviendraient valides plus tard, cependant, ceci peut réduire l'impact de rafales de perte en livrant plus de paquets à l'application. En particulier, un point d'extrémité PEUT conserver les paquets à séquence invalide pour un maximum de deux temps d'aller-retour. Si, dans ce délai, la fenêtre de séquence pertinente change de telle sorte que les paquets deviennent valides en séquence, le point d'extrémité PEUT à nouveau les traiter.

Noter que les paquets DCCP-Reset à séquence invalide provoquent la génération de DCCP-Sync. C'est parce que les points d'extrémité dans un état non synchronisé (CLOSE, DEMANDE, et LISTEN) pourraient ne pas avoir suffisamment d'informations pour générer un bon DCCP-Reset au premier essai. Par exemple, si un point d'extrémité homologue est à l'état CLOSE et reçoit un paquet DCCP-Données, il ne peut pas deviner le bon numéro de séquence à utiliser sur le DCCP-Reset qu'il génère (car le paquet DCCP-Données n'a pas de numéro d'accusé de réception. Le DCCP-Sync généré en réponse à cette mauvaise réinitialisation sert de défi, et contient suffisamment d'informations pour que l'homologue génère un DCCP-Reset approprié. Cependant, le nouveau DCCP-Reset peut porter un autre code de réinitialisation que le DCCP-Reset original ; probablement le nouveau code de réinitialisation sera de 3, "Pas de connexion". Le point d'extrémité DEVRAIT utiliser, si possible, les renseignements provenant du DCCP-Reset original.

7.5.5 Attaques de numéro de séquence

Numéros de séquence et d'accusé de réception forment la principale ligne de défense de DCCP contre les attaquants. Un attaquant qui ne peut pas deviner les numéros de séquence ne peut pas facilement manipuler ou détourner une connexion DCCP, et des exigences comme le choix prudent du numéro de séquence initial éliminent les attaques les plus graves.

Un attaquant peut cependant encore envoyer de nombreux paquets avec des numéros de séquence et d'accusé de réception choisis au hasard. Si un de ces essais se trouve valide en séquence, il peut fermer la connexion ou causer d'autres problèmes. Les plus simples de ces attaques sont exécutées comme suit :

- o Envoyer des paquets DCCP-Données avec des numéros de séquence aléatoires. Si l'un de ces paquets entre dans la fenêtre de séquence de numéros valides, les données d'application du paquet d'attaque peuvent être insérées dans le flux de données.
- o Envoyer des paquets DCCP-Sync paquets avec des numéros de séquence et d'accusé de réception aléatoires. Si l'un de ces paquets entre dans la fenêtre des numéros d'accusé de réception valides, le receveur va par conséquent faire glisser sa fenêtre de numéros de séquence, la faisant sortir de la synchronisation avec le point d'extrémité correct -- peut-être définitivement.

L'attaquant a à deviner les deux accès de source et de destination pour que ces attaques réussissent. En outre, la connexion devrait être inactive pour que l'attaque de DCCP-Sync réussisse, en supposant que le la victime ait mis en œuvre les contrôles les plus rigoureux pour les connexions actives recommandées au paragraphe 7.5.3.

Afin de quantifier la probabilité de succès, soit N le nombre de paquets d'attaque que l'attaquant est prêt à envoyer, W la largeur pertinente de la fenêtre de séquence, et L la longueur des numéros de séquence (24 ou 48). La meilleure stratégie de l'attaquant est d'espacer les paquets de l'attaque de façon uniforme sur l'espace de séquence. Alors, la probabilité de tomber sur une fenêtre de numéros de séquence est P = WN/2^L.
La probabilité de réussite d'une attaque de DCCP-Données à l'aide de courts numéros de séquences égale donc P = WN/2^24. Pour W = 100, l'attaquant doit envoyer plus de 83 000 paquets pour atteindre une probabilité de 50 % de

succès. Pour référence, la plus facile attaque de TCP -- l'envoi d'un SYN avec un numéro de séquence aléatoire, ce qui entraînera une réinitialisation de connexion si il s'inscrit dans la fenêtre -- avec W = 8760 (un chiffre par défaut commun) et L = 32 exige plus de 245 000 paquets pour atteindre une probabilité de 50 % de succès.

Le W d'une connexion rapide sera généralement élevé, augmentant la probabilité de succès de l'attaque pour un N fixé. Si cette probabilité devient inconfortablement élevée avec L = 24, le point d'extrémité DEVRAIT empêcher l'utilisation des numéros de séquence courts en manipulant la caractéristique Autoriser les numéros de séquence courts (voir le paragraphe 7.6.1). La limite de probabilité dépend cependant de l'application. Certaines applications, telles que celles déjà conçues pour gérer la corruption, sont assez résistantes aux attaques par injection de données.
L'attaque DCCP-Sync a L = 48, car les paquets DCCP-Sync utilisent exclusivement des longs numéros de séquence ; en outre, la probabilité de succès est réduite de moitié, puisque seulement la moitié de l'espace de numéro de séquence est valide. Les attaques ont une probabilité de succès réduite d'autant. Pour un grand W de 2000 paquets, l'attaquant doit alors envoyer plus de 10^11 paquets pour atteindre 50 % de chances de succès.

Les attaques impliquant des paquets DCCP-Ack, DCCP-DataAck, DCCP-CloseReq, DCCP-Close, et DCCP-Reset sont plus difficiles, puisque les numéros de séquence et d'accusé de réception doivent tous deux être devinés. La probabilité de succès de l'attaque pour ces types de paquets égale P = WXN/2^(2L), où W est la fenêtre de numéro de séquence, X est la fenêtre de numéro d'accusé de réception, et N et L sont comme précédemment.
Comme les attaques de DCCP-Données avec des numéros de séquence courts sont relativement faciles à exécuter pour les attaquants, DCCP a été conçu pour empêcher ces attaques de commettre des escalades de réinitialisation de connexion ou d'autres conséquences graves. En particulier, toutes les options dont le traitement pourrait provoquer la réinitialisation de la connexion sont ignorées quand elles apparaissent sur des paquets DCCP-Données.

7.5.6 Exemples de traitement des numéros de séquence

Dans l'exemple suivant, DCCP A et DCCP B récupèrent d'une grande salve de pertes qui amène les numéros de séquence de DCCP A hors de la fenêtre de numéros de séquence appropriée de DCCP B.

```
DCCP A                                                  DCCP B
(GSS=1, GSR=10)                                         (GSS=10,GSR=1)
                  -->   DCCP-Data(seq 2)      XXX
                                  ...
                  -->   DCCP-Data(seq 100)    XXX
                  -->   DCCP-Data(seq 101)              --> ???
                                                        seqno hors gamme ; envoyer Sync
OK                <--   DCCP-Sync(seq 11, ack 101)      <--
                                                        (GSS=11, GSR=1)
                  -->   DCCP-SyncAck(seq 102, ack 11)   -->   OK
(GSS=102, GSR=11)                                       (GSS=11, GSR=102)
```

Dans le prochain exemple, une connexion DCCP récupère d'une attaque aveugle simple.

```
DCCP A                                                  DCCP B
(GSS=1,GSR=10)                                          (GSS=10,GSR=1)
                  *ATTACKER*  -->  DCCP-Data(seq 10^6)  --> ???
                                                        seqno hors gamme ; envoyer Sync
   ???                        <--  DCCP-Sync(seq 11, ack 10^6)  <--
ackno hors gamme ; ignorer
(GSS=1,GSR=10)                                          (GSS=11,GSR=1)
```

Le dernier exemple montre la récupération à partir d'une connexion semi-ouverte.

```
DCCP A                                                  DCCP B
(GSS=1,GSR=10                                           (GSS=10,GSR=1)
(Crash)
CLOS                                                        OPEN
REQUEST          -->   Demande-DCCP(seq 400)      -->       ???
  !!             <--   DCCP-Sync(seq 11, ack 400) <--       OUVERT
REQUEST          -->   DCCP-Reset(seq 401, ack 11) -->      (Interrompre)
REQUEST                                                     CLOS
REQUEST          -->   Demande-DCCP(seq 402)      -->       ...
```

7.6 Numéros de séquence courts

Les numéros de séquence DCCP sont longs de 48 bits. Cet espace de grande séquence protège les connexions DCCP contre certaines attaques aveugles, telles que l'injection de DCCP-Reset dans la connexion. Toutefois, les paquets DCCP-Données, DCCP-Ack et DCCP-DataAck, qui constituent le corps de toute connexion DCCP, peuvent réduire l'espace d'en-tête en transmettant uniquement les 24 bits de moindre poids du numéro de séquence et d'accusé de réception. Le point d'extrémité receveur étendra ces numéros à 48 bits en utilisant le pseudo code suivant :

```
Procédure de Extend_Sequence_Number (S, REF)
    /* S est un numéro de séquence de 24 bits provenant de l'en-tête de paquet.
       REF est le numéro de séquence de référence de 48 bits : GSS si S est un numéro d'accusé de réception, et GSR si S
        est un numéro de séquence. */
    Régler REF_low = 24 bits de moindre poids de REF
    Régler REF_hi = 24 bits de poids fort de REF
    Si REF_low (<) S          /* comparaison circulaire mod 2^24 */
         et S |<| REF_low,    /* comparaison conventionnelle non circulaire */
      Retourner (((REF_hi + 1) mod 2^24) << 24) | S
    Sinon, si S (<) REF_low et REF_low |<| S,
      Retourner (((REF_hi - 1) mod 2^24) << 24) | S
    Sinon,
      Retourner (REF_hi << 24) | S
```

Les deux types de comparaison dans les déclarations "si" détectent quand les bits de moindre poids de l'espace des séquences sont revenus à zéro. (La comparaison circulaire "REF_low (<) S" renvoie vrai si et seulement si (S - REF_low), calculé en utilisant l'arithmétique de complément à deux et ensuite représenté comme un nombre non signé, est inférieur ou égal à 2^23 (mod 2^24).) Quand cela arrive, les bits de poids fort sont incrémentés ou décrémentés, comme approprié.

7.6.1 Autoriser la caractéristique Numéros de séquence courts

Les points d'extrémité peuvent exiger que tous les paquets utilisent des numéros de séquence longs en laissant la caractéristique Autoriser les numéros de séquence courts à sa valeur par défaut de zéro. Cela peut réduire le risque que les données soient injectées de façon inappropriée dans la connexion. Le DCCP A envoie une option "Change L (Autoriser les Seqnos courts , 1)" pour indiquer sa volonté d'envoyer des paquets avec des numéros de séquence courts.

Autoriser les numéros de séquence courts a le numéro de caractéristique 2 et est une priorité de serveur. Cela prend des valeurs booléennes d'un octet. Lorsque Autoriser les Seqnos courts/B est zéro, le DCCP B NE DOIT PAS envoyer des paquets avec de courts numéros de séquences et le DCCP A DOIT ignorer tous les paquets qui sont reçus avec des numéros de séquences courts. Les valeurs de deux ou plus sont réservées. Les nouvelles connexions commencent par Autoriser les numéros de séquence courts 0 pour les deux points d'extrémité.

7.6.2 Quand éviter les numéros de séquence courts

Les numéros de séquence courts réduisent le débit que les connexions DCCP peuvent atteindre en toute sécurité et augmentent les risques de certains types d'attaques, y compris l'injection aveugle de données. Les connexions DCCP à très haut débit, et celles avec de grandes fenêtres de numéros de séquence (paragraphe 7.5.2), NE DEVRAIENT PAS utiliser des numéros de séquence courts sur leurs paquets de données. Les questions de risque d'attaque ont été discutées au paragraphe 7.5.5, on discute ici de la limitation de débit.

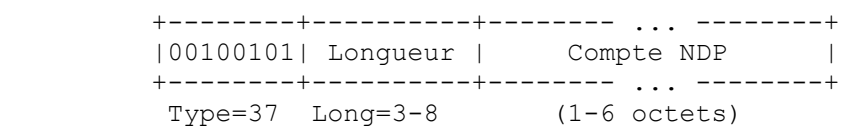
Le mécanisme de validité de séquence suppose que le réseau ne livre pas des données extrêmement vieilles. En particulier, il suppose que le réseau doit avoir abandonné un paquet au moment où la connexion fait revenir à zéro ses numéros de séquence et utilise son numéro de séquence à nouveau. Cette contrainte limite le taux de connexion maximum qui peut être atteint en toute sécurité. Soit MSL la durée de vie maximale d'un segment, P la taille moyenne en bits d'un paquet DCCP, et L la longueur des numéros de séquence (24 ou 48 bits). Le taux maximum sûr, en bits par seconde, est R = P*(2^L)/2MSL.

Pour le MSL par défaut de 2 minutes, des paquets DCCP de 1500 octets et à courts numéros de séquence, le taux sûr est donc d'environ 800 Mbit/s. Bien que 2 minutes soit un très grand MSL pour tous les réseaux qui pourraient tenir ce taux avec de si petits paquets, les numéros de séquence longs vont permettre des taux beaucoup plus élevés sous les mêmes contraintes : jusqu'à 14 petabits par seconde pour des paquets de 1500 octets et le MSL par défaut.

7.7 Compte NPD et détection de perte d'application

Les numéros de séquence DCCP s'incrémentent de un à chaque paquet, y compris les paquets qui ne sont pas de données

(paquets qui ne portent pas de données d'application). Cette rend les numéros de séquence DCCP adaptés pour détecter toute perte dans le réseau, mais pas pour la détection de pertes de données d'application. L'option Compte NPD rapporte la longueur de chaque salve de paquets non de données. Cela permet au receveur DCCP de déterminer avec fiabilité quand une rafale de pertes inclut des données des applications.



Si la caractéristique Envoyer le compte NDP d'un point d'extrémité DCCP est un (voir ci-dessous) ce point d'extrémité DOIT alors envoyer une option Compte NPD sur chaque paquet dont le prédécesseur immédiat était un paquet non de données. Les paquets non de données se composent des types de paquets DCCP DCCP-Ack, DCCP-Close, DCCP-CloseReq, DCCP-Reset, DCCP-Sync, et DCCP-SyncAck. Les autres types de paquets, à savoir Demande-DCCP, DCCP-Réponse, DCCP-Données, et DCCP-DataAck, sont considérés comme des paquets de données, bien que tous les paquets Demande-DCCP et DCCP-Réponse ne transportent pas, en fait, de données d'application.

La valeur mémorisée dans le compte NPD est égale au nombre de paquets de données non consécutifs dans la plage précédant immédiatement le paquet actuel. Les paquets sans option Compte NPD sont considérés comme ayant le compte NPD zéro.

L'option Compte NPD peut porter de un à six octets de données. Le plus petit format de l'option qui peut contenir le compte NPD DEVRAIT être utilisé.

Avec le compte NPD, le receveur peut dire de manière fiable uniquement si une rafale de pertes contenaient au moins un paquet de données. Par exemple, le receveur ne peut pas toujours dire si une rafale de pertes contenait un paquet non de données.

7.7.1 Notes sur l'utilisation du compte NDP

Disons que K numéros de séquences consécutifs sont manquants dans une certaine salve de pertes, et que la caractéristique Envoyer compte NPD est activée. Ensuite, certaines données d'application ont été perdues au sein de ces numéros de séquence à moins que le paquet suivant le trou contienne une option Compte NPD dont la valeur est supérieure ou égale à K.

Par exemple, disons qu'un point d'extrémité a envoyé la séquence suivante de paquets non de données (Nx) et de paquets de données (Dx) :

N0 N1 D2 N3 D4 D5 N6 D7 D8 D9 D10 N11 N12 D13

Ces paquets auraient les comptes NPD suivants :

N0	N1	D2	N3	D4	D5	N6	D7	D8	D9	D10	N11	N12	D13
-	1	2	-	1	-	-	1	-	-	-	-	1	2

Le compte NPD n'est pas utile pour des applications qui incluent leurs propres numéros de séquence dans leurs en-têtes de paquets.

7.7.2 Caractéristique Envoyer le compte NPD

La caractéristique Envoyer le compte NPD permet aux points d'extrémité DCCP de négocier s'ils devraient envoyer des options Compte NPD sur leurs paquets. Le DCCP A envoie une option "Change R (Envoyer compte NDP, 1)" pour demander au DCCP B d'envoyer des options Compte NDP.

Envoyer le compte NPD a le numéro de caractéristique 7 et est une priorité de serveur. Elle prend des valeurs booléennes d'un octet. Le DCCP B DOIT envoyer les options Compte NPD comme décrit ci-dessus lorsque Envoyer compte NPD/B est à un, bien qu'il PUISSE envoyer des options Compte NDP même lorsque Envoyer le compte NPD/B est égal à zéro. Les valeurs de deux ou plus sont réservées. Les nouvelles connexions commencent par Envoyer compte NDP 0 pour les deux points d'extrémité.

8. Traitement d'événements

Cette section décrit comment les connexions DCCP changent entre les états et quels paquets sont envoyés et à quel moment. Noter que la négociation de caractéristiques a lieu en parallèle avec les transitions d'état à l'échelle de la connexion décrites ici.

8.1 Établissement de la connexion

La phase d'initialisation des connexions DCCP se compose d'une prise de contact en trois étapes : un paquet de Demande-DCCP initial envoyé par le client, une DCCP-Réponse envoyée par le serveur en réponse, et enfin un accusé de réception du client, généralement via un paquet DCCP-Ack ou DCCP-DataAck. Le client passe de l'état DEMANDE à PARTOPEN, et enfin à OPEN ; le serveur se déplace de LISTEN à RESPOND, et enfin à OPEN.

	État du client				**État du serveur**
	CLOSED				LISTEN
1.	REQUEST	-->	Request	-->	
2.		<--	Response	<--	RESPOND
3.	PARTOPEN	-->	Ack, DataAck	-->	
4		<--	Data, Ack, DataAck	<--	OPEN
5.	OPEN	<->	Data, Ack, DataAck	<->	OPEN

8.1.1 Demande du client

Quand un client décide d'initier une connexion, il entre dans l'état DEMANDE, choisit un numéro de séquence initial (paragraphe 7.2) et envoie au serveur prévu un paquet Demande-DCCP en utilisant ce numéro de séquence.

Les paquets Demande-DCCP vont fréquemment porter des caractéristiques d'options de négociation qui ouvrent des négociations pour divers paramètres de connexion, tels que les identifiants de contrôle d'encombrement préférés pour chaque demi-connexion. Ils peuvent également transporter des données d'application, mais le client doit être conscient que le serveur peut ne pas accepter de telles données.

Un client à l'état DEMANDE DEVRAIT utiliser un temporisateur de retard exponentiel pour envoyer de nouveaux paquets Demande-DCCP si aucune réponse n'est reçue. La première retransmission devrait se produire après environ une seconde, reculant à pas moins d'un paquet toutes les 64 secondes ; ou le point d'extrémité peut utiliser toute stratégie de retransmission suivie pour les SYN TCP retransmettants. Chaque nouvelle Demande-DCCP DOIT incrémenter le numéro de séquence de un et DOIT contenir le même code de service et de données d'application que la Demande-DCCP d'origine.

Un client PEUT renoncer à ses Demande-DCCP après un certain temps (3 minutes, par exemple). Lorsque c'est le cas, il DEVRAIT envoyer un paquet DCCP-Reset au serveur avec le code de réinitialisation 2, "Interrompu", pour nettoyer l'état au cas où plusieurs des demandes seraient en fait arrivées. Un client dans l'état DEMANDE n'a jamais reçu un numéro de séquence initial de ses homologues, de sorte que le numéro d'accusé de réception du DCCP-Reset DOIT être mis à zéro.

Le client quitte l'état de DEMANDE pour PARTOPEN quand il reçoit une DCCP-Réponse du serveur.

8.1.2 Codes de service

Chaque Demande-DCCP contient un code de service de 32 bits, qui identifie le service au niveau application auquel l'application du client essaye de se connecter. Les codes de service devraient correspondre aux services et protocoles d'application. Par exemple, il pourrait y avoir un code de service pour les connexions de contrôle SIP et un autre pour les connexions audio RTP. Les boîtiers de médiation, tels que les pare-feu, peuvent utiliser le code de service pour identifier l'application fonctionnant sur un accès non standard (en supposant que l'en-tête DCCP n'a pas été chiffré).

Les points d'extrémité DOIVENT associer un code de service à toutes les prises DCCP, fonctionnant à la fois activement et passivement. L'application va généralement fournir ce code de service. Chaque prise active DOIT avoir exactement un code de service. Les prises passives PEUVENT, à la discrétion de la mise en œuvre, être associées à plus d'un code de service ; cela pourrait laisser plusieurs applications, ou plusieurs versions de la même application, écouter sur le même accès, différenciées par le code de service. Si le code de service de la Demande-DCCP n'est égal à aucun des codes de service du serveur pour l'accès donné, le serveur DOIT rejeter la demande en envoyant un paquet DCCP-Reset avec le code de réinitialisation 8, "Mauvais code de service". Un boîtier de médiation PEUT aussi envoyer un tel DCCP-Reset en réponse à des paquets dont le code de service est considéré comme inapproprié.

Les codes de service ne sont pas destinés à être spécifiques de DCCP et sont attribués par l'IANA. Suivant les politiques énoncées dans la [RFC2434], la plupart des codes de service sont attribués selon la politique du premier arrivé, premier servi, sous réserve des lignes directrices suivantes :

- Les codes de service sont attribués un à la fois, ou en petits blocs. Une courte description en anglais du service prévu est EXIGÉE pour obtenir une affectation de code de service, mais aucune spécification, en cours de normalisation ou autre, n'est nécessaire. L'IANA gère l'association des codes de service aux phrases correspondantes.

- Les utilisateurs demandent des valeurs de code de service spécifiques. On suggère que les utilisateurs demandent des codes de service qui peuvent être représentés en utilisant la convention de formatage "SC:" décrite ci-dessous. Ainsi, le "Protocole Frobodyne Plotz" pourrait correspondre au code de service 17178548426, ou, de façon équivalente, à "CS:fdpz". L'interprétation canonique d'un champ de code de service est numérique.

- Les codes de service dont les octets ont chacun des valeurs dans l'ensemble {32, 45-57, 65-90} utilisent une politique d'allocation par spécification exigée. C'est-à-dire que ces codes de service sont utilisés pour une norme internationale ou des spécifications en voie de normalisation, de l'IETF ou autres. (Cet ensemble se compose des chiffres, des lettres majuscules et caractères d'espace, "-", "."et "/" ASCII).

- Les codes de service dont l'octet de poids fort est égal à 63 (ASCII "?") sont réservés à un usage privé.

- Le code de service 0 représente l'absence d'un code de service significatif et NE DOIT PAS être attribué.

- La valeur 4 294 967 295 est un code de service non valide. Les serveurs DOIVENT rejeter toute Demande-DCCP avec cette valeur de code de service en envoyant un paquet DCCP-Reset avec le code de réinitialisation de 8, "Mauvais code de service".

Cette conception de l'attribution des codes de service se fonde sur l'allocation des identifiants de quatre octets pour les ressources Macintosh, les tronçons de PNG, et les tableaux TrueType et OpenType.

Dans les réglages de texte, on recommande que les codes de service soient rédigés sous une des trois formes suivantes, préfixés par les lettres ASCII SC et, soit deux points ":" , soit le signe égal "=". Ces formes sont interprétées comme suit.

SC : Indique un code de service représentable en utilisant un sous ensemble des caractères ASCII. Le caractère "deux points" est suivi par un à quatre caractères tirés de l'ensemble suivant : lettres, chiffres, et les caractères "-_+.*/?@" (non compris les guillemets). Numériquement, ces caractères ont des valeurs dans {42-43, 45-57, 63-90, 95, 97-122}. Le code de service est calculé en bourrant la chaîne sur la droite avec des espaces (valeur 32) et en interprétant le résultat à quatre caractères comme un nombre gros boutien de 32 bits.

SC = : Indique un code de service décimal. Le signe égal est suivi par n'importe quel nombre de chiffres décimaux, qui précisent le code de service. Les valeurs supérieures à 4 294 967 294 sont illégales.

SC = x ou SC = X : Indique un code de service hexadécimal. Le "x" ou "X" est suivi par un nombre quelconque de chiffres hexadécimaux (en majuscules ou minuscules) qui précisent le code de service. Les valeurs supérieures à 4 294 967 294 sont illégales.

Ainsi, le Code du service 1717858426 peut être représentée en texte par SC: fdpz, SC = 1717858426, ou SC = x6664707A.

8.1.3 Réponse du serveur

Dans la deuxième phase de la prise de contact à trois phases, le serveur passe de l'état LISTEN à l'état RESPONSE et envoie un message DCCP-Réponse au client. Dans cette phase, un serveur va souvent spécifier les fonctionnalités qu'il aimerait utiliser, que ce soit parmi celles demandées par le client ou en plus de celles-ci. Parmi ces options il y a le mécanisme de contrôle d'encombrement que le serveur s'attend à utiliser.

Le serveur PEUT répondre à un paquet Demande-DCCP avec un paquet DCCP-Reset pour refuser la connexion. Les codes pertinents de réinitialisation pour refuser une connexion comprennent : 7, "Connexion refusée", lorsque l'accès de destination de la Demande-DCCP ne correspondait pas à un accès DCCP ouvert pour l'écoute ; 8, "Mauvais code de service", lorsque le code de service de la Demande-DCCP ne correspond pas au code de service enregistré auprès de l'accès de destination, et 9, "Trop occupé", lorsque le serveur est actuellement trop occupé pour répondre aux demandes. Le serveur DEVRAIT limiter le débit auquel il génère ces réinitialisations, par exemple, à pas plus de 1024 par seconde.

Le serveur NE DEVRAIT PAS retransmettre les paquets DCCP-Réponse ; le client va retransmettre la Demande-DCCP si

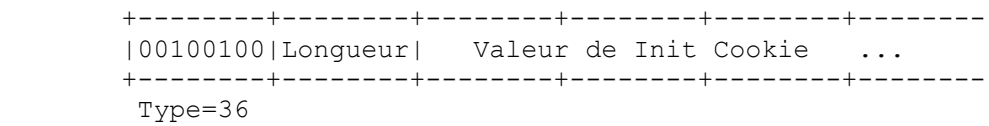
nécessaire. (Noter que la Demande-DCCP "Retransmis" aura, au moins, un autre numéro de séquence que la Demande-DCCP '"originale". Le serveur peut ainsi distinguer la vraie retransmission des dupliqués par le réseau.) Le serveur va détecter que la Demande-DCCP retransmise s'applique à une connexion existante en raison de ses accès de source et de destination. Chaque Demande-DCCP valide reçue alors que le serveur est dans l'état RESPOND DOIT susciter une nouvelle DCCP-Réponse. Chaque nouvelle DCCP-Réponse DOIT incrémenter de un le numéro de séquence du serveur et DOIT inclure les mêmes données d'application, s'il en est, que la DCCP-Réponse d'origine.

Le serveur NE DOIT PAS accepter plus d'un tronçon de données d'application de Demande-DCCP par connexion. En particulier, le DCCP-Réponse envoyé en réponse à une Demande-DCCP retransmise avec des données d'application DEVRAIT contenir une option Données éliminées, dans laquelle les données retransmises de la Demande-DCCP sont signalées avec le code d'abandon 0, Contraintes du protocole. La Demande-DCCP d'origine DEVRAIT également être signalée dans l'option Données éliminées, soit dans un bloc normal (si le serveur a accepté les données ou si il n'y avait pas de données) soit dans un code d'abandon 0, Bloc abandonné (si le serveur avait refusé les données aussi la première fois).

Les options Données éliminées et Init Cookie sont particulièrement utiles pour les paquets DCCP-Réponse (paragraphes 11.7 et 8.1.4).

Le serveur quitte l'état RESPOND pour OPEN quand il reçoit un DCCP-Ack valide du client, terminant la prise de contact à trois phases. Il PEUT également quitter l'état RESPOND pour l'état CLOSED après un délai d'au moins que 4MSL (8 minutes) ; lorsque il le fait, il DEVRAIT envoyer une DCCP-Reset avec le code de réinitialisation 2, "Interrompu", pour nettoyer l'état au niveau du client.

8.1.4 Option Init Cookie



L'option Init Cookie permet à un serveur DCCP éviter d'avoir à détenir un état jusqu'à l'achèvement de la prise de contact à trois phases d'établissement de connexion, de manière similaire à celle des mouchards SYN TCP [SYNCOOKIES]. Le serveur enveloppe le code de service, l'accès du serveur, et toutes les options auxquelles il tient à la fois de la Demande-DCCP et de la DCCP-Réponse dans un mouchard opaque. Généralement, le mouchard sera chiffré à l'aide un secret connu seulement du serveur et va comporter une somme de contrôle cryptographique ou une valeur "magique" afin de vérifier que le déchiffrement est correct. Lorsque le serveur reçoit le mouchard dans la réponse, il peut le déchiffrer et instancier tout l'état qu'il a évité de garder. Dans l'intervalle, il n'a pas besoin de quitter l'état LISTEN.

L'option Init Cookie NE DOIT PAS être envoyée sur les paquets Demande-DCCP ou DCCP-Données. Toutes les options Init Cookie reçues sur des paquets Demande-DCCP ou DCCP-Données, ou après que la connexion a été établie (lorsque l'état de la connexion est ? OPEN), DOIVENT être ignorées. Le serveur PEUT inclure des options Init Cookie dans sa DCCP-Réponse. Si il en est ainsi, le client DOIT alors faire écho aux mêmes options Init Cookie, dans le même ordre, dans chaque paquet DCCP suivant jusqu'à ce qu'un de ces paquets soit acquitté (montrant que la prise de contact à trois phases est terminée) ou que la connexion soit réinitialisée. En conséquence, le client NE DOIT PAS utiliser les paquets DCCP-Données jusqu'à ce que la prise de contact à trois phases soit achevée ou que la connexion soit réinitialisée. Les options Init Cookie sur un paquet client DOIVENT être égales à celles reçues sur le Demande-DCCP indiqué par le numéro d'accusé de réception du paquet du client. Le serveur DEVRAIT concevoir le format de ses Init Cookie de telle sorte que l'intégrité des Init Cookie puisse être vérifiée ; il DEVRAIT répondre à une option Init Cookie altérée en réinitialisant le connexion avec le code de réinitialisation 10, "Mauvais Init Cookie".

La mise en œuvre précise des Init Cookie n'a pas à être spécifiée ici ; comme les Init Cookie sont opaques pour le client, il n'y a pas de problème d'interopérabilité. Un exemple de format de mouchard pourrait chiffrer (en utilisant une clé secrète) la séquence initiale de la connexion et les numéros d'accusé de réception, les accès, le code de service, toutes les options incluses dans le paquet Demande-DCCP et la DCCP-Réponse correspondante, un sel aléatoire, et un numéro magique. À la réception d'un Init Cookie reflété, le serveur va déchiffrer le mouchard, le valider en vérifiant son numéro magique, son numéro de séquence, et les accès, et, s'il est valide, créer une prise correspondante en utilisant les options.

Chaque option Init Cookie individuelle peut contenir au maximum 253 octets de données, mais un serveur peut envoyer de multiples options de Init Cookie pour gagner plus d'espace.

8.1.5 Achèvement de la prise de contact

Lorsque le client reçoit une DCCP-Réponse du serveur, il passe de l'état REQUEST à l'état PARTOPEN et achève la prise

de contact à trois phases en envoyant un paquet DCCP-Ack au serveur. Le client reste dans l'état PARTOPEN jusqu'à ce qu'il puisse être sûr que le serveur a reçu des paquets envoyés par le client à partir de l'état PARTOPEN (soit le DCCP-Ack initial, soit un paquet ultérieur). Les clients dans l'état PARTOPEN qui veulent envoyer des données DOIVENT le faire en utilisant des paquets DCCP-DataAck, pas des paquets DCCP-Données. C'est parce que les paquets DCCP-Données n'ont pas de numéro d'accusé de réception, de sorte que le serveur ne peut pas dire à partir d'un paquet DCCP-Données si le client a vu sa DCCP-Réponse. Par ailleurs, si le DCCP-Réponse incluait un Init Cookie, ce Init Cookie DOIT être inclus dans tous les paquets envoyés dans l'état PARTOPEN.

Le seul DCCP-Ack envoyé lors de l'entrée dans l'état PARTOPEN pourrait, bien sûr, être abandonné par le réseau. Le client DEVRAIT s'assurer que certains paquets passent finalement. Le mécanisme préféré serait un temporisateur d'à peu près 200 millisecondes, établi chaque fois qu'un paquet est transmis en PARTOPEN. Si ce temporisateur arrive à expiration et que le client est encore en PARTOPEN, le client génère un autre DCCP-Ack et ajoute un retard à croissance exponentielle à la minuterie. Si le client reste dans PARTOPEN plus de 4MSL (8 minutes), il faut réinitialiser la connexion avec le code Réinitialiser 2, "Interrompu".

Le client quitte l'état PARTOPEN pour OPEN quand il reçoit un paquet valide autre que DCCP-Réponse, DCCP-Reset ou DCCP-Sync du serveur.

8.2 Transfert de données

Dans la phase de transfert de données centrale de la connexion, le serveur et le client sont tous deux dans l'état Ouvert.

Le DCCP A envoie des paquets DCCP-Données et DCCP-DataAck au DCCP B en raison des événements d'application sur l'hôte A. Ces paquets sont contrôlés pour l'encombrement par le CCID pour la demi-connexion de A à B. En revanche, les paquets DCCP-Ack envoyés par le DCCP A sont contrôlés par le CCID pour la demi-connexion B à A. Généralement, le DCCP A va porter les informations d'accusé de réception sur des paquets DCCP-Données lorsque ils sont acceptables, créant des paquets DCCP-DataAck. Les paquets DCCP-Ack sont utilisés quand il y n'existe pas de données à envoyer de DCCP A à DCCP B, ou lorsque l'état d'encombrement du CCID de A à B ne permet pas d'envoyer de données.

Les paquets DCCP-Sync et DCCP-SyncAck peuvent aussi se produire dans la phase de transfert des données. Certains cas qui causent la génération de DCCP-Sync sont discutés au paragraphe 7.5. Une distinction importante entre les paquets DCCP-Sync et les autres types de paquets est que DCCP-Sync provoque un accusé de réception immédiat. À réception d'un paquet DCCP-Sync valide, un point d'extrémité DCCP DOIT immédiatement générer et envoyer une réponse DCCP-SyncAck (sous réserve de toute limite de taux d'exécution) ; le numéro d'accusé de réception sur ce DCCP-SyncAck DOIT être égal au numéro de séquence du DCCP-Sync.

Une mise en œuvre particulière de DCCP pourrait décider de n'initier de négociation de caractéristique qu'une fois que l'état OPEN a été atteint, auquel cas il pourrait ne pas permettre le transfert de données pendant un certain temps. Les données reçues pendant ce temps DEVRAIENT être rejetées et rapportées en utilisant un bloc d'abandon Données abandonnées avec le code d'abandon 0, Contraintes de protocole (voir au paragraphe 11.7).

8.3 Terminaison

La terminaison de connexion DCCP utilise une prise de contact composée d'un paquet DCCP-CloseReq facultatif, d'un paquet DCCP-Close, et d'un DCCP-Reset. Le serveur passe de l'état OUVERT, éventuellement par l'état CLOSEREQ, à l'état FERMÉ, le client passe de OPEN à CLOSING puis à TIMEWAIT, et après le temps d'attente 2MSL (4 minutes) à FERMÉ.

La séquence DCCP-CloseReq, DCCP-Close, DCCP-Reset est utilisée lorsque le serveur décide de fermer la connexion, mais ne veut pas tenir l'état TIMEWAIT :

État du client				**État du serveur**
OPEN				OPEN
1.	<--	CloseReq	<--	CLOSEREQ
2. CLOSING	-->	Close	-->	
3.	<--	Reset	<--	CLOSED (LISTEN)
4. TIMEWAIT				
5. CLOSED				

Une plus courte séquence se produit lorsque le client décide de fermer la connexion.

État du client				**État du serveur**
OPEN				OPEN
1. CLOSING	-->	Close	-->	
2.	<--	Reset	<--	CLOSED (LISTEN)
3. TIMEWAIT				
4. CLOSED				

Enfin, le serveur peut décider de tenir l'état TIMEWAIT :

État du client				**État du serveur**
OPEN				OPEN
1.	<--	Close	<--	CLOSING
2. CLOSED	-->	Reset	-->	
3.				TIMEWAIT
4.				CLOSED (LISTEN)

Dans tous les cas, le receveur du paquet DCCP-Reset conserve l'état TIMEWAIT POUR la connexion. Comme dans le protocole TCP, l'état TIMEWAIT, où un point d'extrémité préserve discrètement une prise pendant 2MSL (4 minutes) après la fermeture de sa connexion, garantit qu'aucune connexion dupliquant les adresses et les accès de source et destination de la connexion en cours ne peut démarrer tant que d'anciens paquets peuvent rester dans le réseau.

La prise de contact de résiliation procède comme suit. Le receveur d'un paquet DCCP-CloseReq valide DOIT répondre avec un paquet DCCP-Close. Le receveur d'un paquet DCCP-Close valide DOIT répondre par un paquet DCCP-Reset avec le code de réinitialisation 1, "Fermé". Le receveur d'un paquet DCCP-Reset valide -- qui est aussi l'envoyeur du paquet DCCP-Close (et éventuellement le receveur du paquet DCCP-CloseReq) -- conservera l'état TIMEWAIT pour la connexion.

Un paquet DCCP-Reset achève chaque connexion DCCP, que la terminaison soit propre (en raison de la fermeture de l'application ; code de réinitialisation 1, "Fermé") ou pas. Contrairement à TCP, qui a deux mécanismes distincts de terminaison (FIN et RST) DCCP termine toutes les connexions d'une manière uniforme. Cela se justifie parce que certains aspects de la terminaison de la connexion sont les mêmes indépendamment de savoir si la terminaison a été propre. Par exemple, le point d'extrémité qui reçoit un DCCP-Reset valide DEVRAIT tenir l'état TIMEWAIT pour la connexion. Les processeurs qui doivent faire la distinction entre la terminaison propre et non propre peuvent examiner le code de réinitialisation. Les mises en œuvre DCCP passent généralement à l'état FERMÉ après l'envoi d'un paquet DCCP-Reset.

Les points d'extrémité dans les états CLOSEREQ et CLOSING DOIVENT retransmettre les paquets, respectivement, DCCP-CloseReq et DCCP-Close jusqu'à ce qu'ils quittent ces états. Le temporisateur de retransmission devrait d'abord être réglé à se déclencher dans deux temps d'aller-retour et devrait reculer d'au moins une fois toutes les 64 secondes si aucune réponse pertinente n'est reçue.

Seul le serveur peut envoyer un paquet DCCP-CloseReq ou entrer dans l'état CLOSEREQ. Un serveur qui reçoit un paquet DCCP-CloseReq dont le numéro de séquence est valide DOIT répondre avec un paquet DCCP-Sync et autrement ignorer le DCCP CloseReq.

Les paquets DCCP-Données, DCCP-DataAck et DCCP-Ack reçus dans les états CLOSEREQ ou CLOSING PEUVENT être traités ou ignorés.

8.3.1 Terminaison anormale

Les points d'extrémité DCCP génèrent des paquets DCCP-Reset pour terminer anormalement les connexions ; un paquet DCCP-Reset peut être généré à partir de n'importe quel état. Les Reset envoyés dans les états FERMÉ, ÉCOUTE et TIMEWAIT utilisent le code de réinitialisation 3, "Pas de connexion", sauf indication contraire. Les Reset envoyés dans les états DEMANDE ou RÉPONSE utilisent le code de réinitialisation 4, "Erreur de paquet", sauf indication contraire.

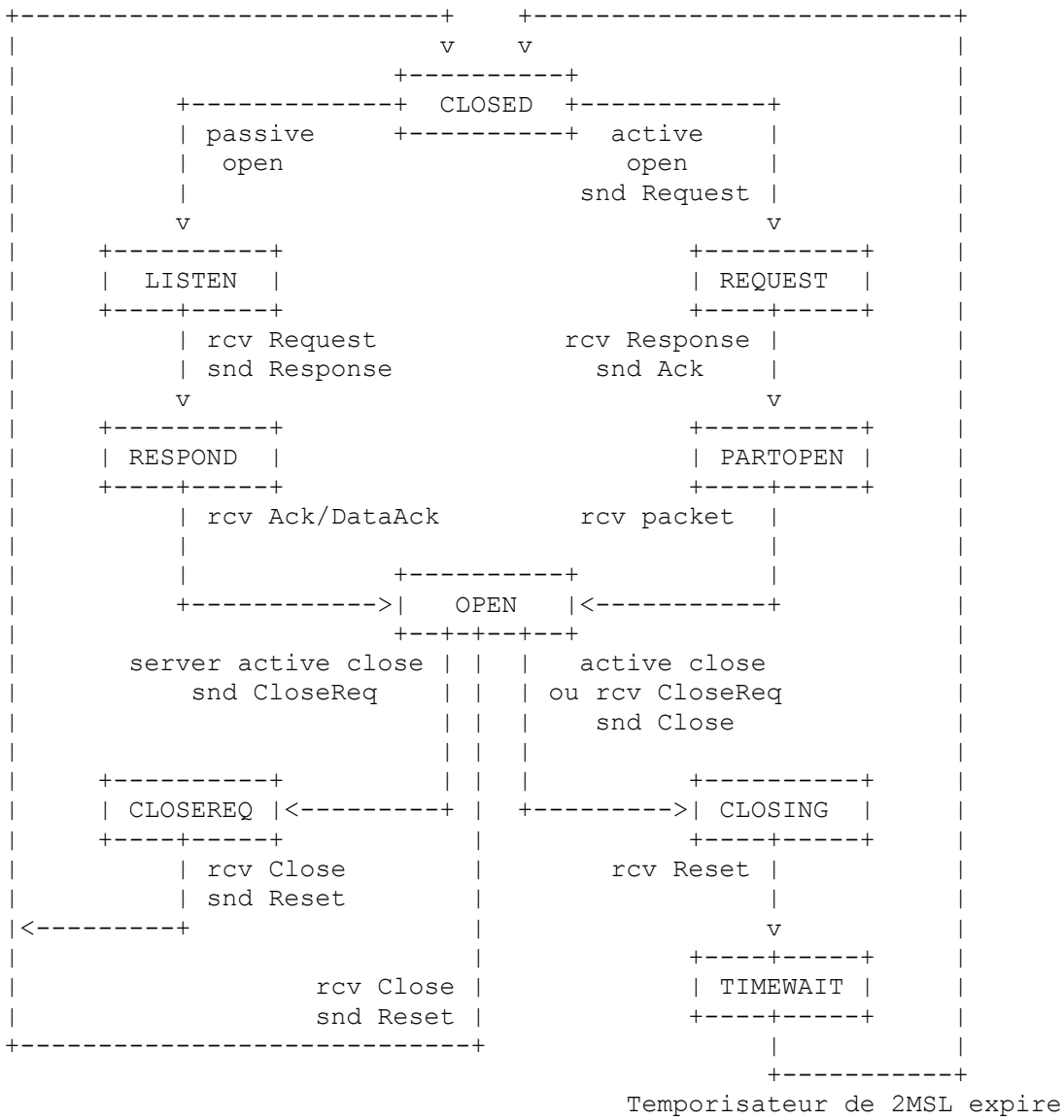
Les points d'extrémité DCCP dans les états FERMÉ, ÉCOUTER, ou TIMEWAIT peuvent avoir besoin de générer un paquet DCCP-Reset en réponse à un paquet reçu d'un homologue. Comme ces états n'ont pas de variable de numéro de séquence associée, les numéros de séquence et d'accusé de réception sur le paquet DCCP-Reset R sont tirés du paquet P reçu, comme suit.

1. Si P.ackno existe, régler alors R.seqno: = P.ackno + 1. Sinon, régler R.seqno: = 0.
2. Régler R.ackno: = P.seqno.
3. Si le paquet utilise des numéros de séquence courts (PX == 0), régler alors les 24 bits de poids fort de R.seqno et

R.ackno à 0.

8.4 Diagramme d'état DCCP

Les transitions d'état les plus courantes discutées ci-dessus peuvent être résumées dans le diagramme d'état suivant. Le schéma est une illustration ; le texte du paragraphe 8.5 et ailleurs devrait être considéré comme définitif. Par exemple, il y a des arcs (non illustrés) à partir de chaque état, sauf CLOSED à TIMEWAIT, subordonnés à la réception d'un DCCP-Reset valide.



8.5 Pseudo code

Cette section présente un algorithme décrivant les étapes de traitement qu'un point d'extrémité DCCP DOIT suivre quand il reçoit un paquet. Une mise en œuvre DCCP n'a pas besoin de mettre en œuvre l'algorithme comme il est décrit ici, mais toute mise en œuvre DOIT générer des effets observables exactement comme indiqué par ce pseudo-code, sauf si autorisé autrement par une autre partie de ce document.

Le paquet reçu est décrit par P, la prise par S. Les variables de prises sont :
S.SWL - fenêtre de numéro de séquence basse
S.SWH - fenêtre de numéro de séquence haute
S.AWL - fenêtre de numéro d'accusé de réception basse
S.AWH - fenêtre de numéro d'accusé de réception haute
S.ISS - numéro de séquence initial envoyé
S.ISR - numéro de séquence initial reçu

S.OSR - premier numéro de séquence reçu OUVERT
S.ESG - plus grand nombre de séquence envoyé
S.GSR - plus grand numéro de séquence valide reçu
S.GAR - plus grand numéro d'accusé de réception valide reçu sur une non-Sync ; initialisé à S.ISS
Les actions "Envoyer des paquets" utilisent toujours, et incrémentent, S.GSS.

Étape 1 : Vérifier les éléments de base d'en-tête
/* Cette étape vérifie les paquets malformés. Les paquets qui échouent à ces contrôles sont ignorés - ils ne reçoivent pas de Reset en réponse */
Si le paquet est inférieur à 12 octets, abandon de paquet et retour
Si P.type n'est pas compris, abandon de paquet et retour
Si le décalage de P.Données est plus petit que la longueur fixée pour l'en-tête du type de paquet donné ou supérieur à la longueur du paquet, abandon du paquet et retour
Si P.type n'est pas Données, Ack, ou DataAck et si PX == 0 (le paquet a des numéros de séquence courts) abandon du paquet et retour
Si la somme de contrôle d'en-tête est incorrecte, abandon du paquet et retour
Si P.CsCov est trop grand pour la taille du paquet, abandon du paquet et retour

Étape 2 : Vérifier les accès et traiter l'état TIMEWAIT
/* L'ID de flux est le quadruplet <adresse de source, accès de source, adresse de destination, accès de destination> */
Chercher l'identifiant de flux dans le tableau et obtenir la prise correspondante
Si il n'y a aucune prise, ou si S.state == TIMEWAIT,
/* Les numéros de séquence et d'accusés de réception du Reset suivants sont pris à partir du paquet d'entrée ; voir au paragraphe 8.3.1. */
Générer Reset (Pas de connexion) sauf si P.type == Reset
Abandonner le paquet et retour

Étape 3 : Traitement de l'état LISTEN
Si S.state == ÉCOUTE,
Si P.type == Demande ou si P contient une option valide de Init Cookie
/* Doit examiner les options du paquet pour chercher les Init Cookie. Seuls les Init Cookie sont traités ici ; d'autres options sont traitées dans l'étape 8. Cet examen ne doit être effectué que si le point d'extrémité utilise des Init Cookie */
/* Générer une nouvelle prise et basculer sur cette prise */
Régler S: = nouvelle prise pour cette paire d'accès
S.state = RÉPONDRE
Choisir S.ISS (seqno initial) ou régler à partir des Init Cookie
Initialiser S. GAR: = S. ISS
Régler S.ISR, S.GSR, S.SWL, S.SWH à partir du paquet ou des Init Cookie
Continuer avec S.state == RÉPONDRE
/* Un paquet Réponse va être généré à l'étape 11 */
Sinon,
Générer Reset (Pas de connexion), sauf si P.type == Reset
Abandonner le paquet et retour

Étape 4 : Préparer les numéros de séquence dans DEMANDE
Si S.state == DEMANDE,
Si (P.type == Réponse ou P.type ==Réinitialiser) et S.AWL ? P.ackno ? S. AWH,
/* Régler les variables de numéro de séquence correspondant à l'autre extrémité, donc P va passer les tests de l'étape 6 */
Régler S.GSR, S.ISR, S.SWL, S.SWH
/* Le traitement de la réponse se poursuit à l'étape 10 ; le traitement de Reset continue à l'étape 9 */
Sinon,
/* Seuls Response et Reset sont valables dans l'état Demande */
Générer un Reset (Erreur de paquet)
Abandonner le paquet et retour

Étape 5 : Préparer les numéros de séquence pour la synchronisation
Si P.type == Sync ou P.type == SyncAck,
Si S.AWL ? P.ackno ? S. AWH et P.seqno ? S.SWL,
/* P est valide, dont les variables de numéro de séquence sont mises à jour en conséquence. Après cette mise à jour, P va passer les essais de l'étape 6. Un SyncAck est généré si nécessaire à l'étape 15 */
Mise à jour de S.GSR, S.SWL, S.SWH
Sinon,
Abandonner le paquet et retour

```
Étape 6 : Vérifier les numéros de séquence
    Si PX == 0 et la caractéristique pertinente Autoriser court Seqnos est 0,
/* Le paquet a des numéros de séquence courts, mais ils ne sont pas autorisés */
      Abandonner le paquet et retour
    Sinon, si P.X == 0,
      Étendre P.seqno et P.ackno à 48 bits en utilisant la procédure du paragraphe 7.6
    Soit LSWL = S.SWL et LAWL = S.AWL
    Si, P.type == CloseReq ou P.type == Fermer ou P.type == Réinitialiser
      LSWL: = S. GSR + 1, LAWL: = S. GAR
    Si LSWL ? P.seqno ? S. SWH
        et (P.ackno n'existe pas ou LAWL ? P.ackno ? S. AWH),
      Mettre à jour S.GSR, S.SWL, S.SWH
      Si P.type != Sync,
        Mettre à jour S.GAR
    Sinon,
      Si P.type == Reset,
        Envoyer des paquets Sync accusant réception de S.GSR
      Sinon,
        Envoyer des paquets Sync accusant réception de P.seqno
      Abandonner le paquet et retour

Étape 7: Vérifiez les types de paquets inattendus
    Si (S.is_server et P.type == CloseReq)
       ou (S.is_server et P.type == Réponse)
       ou (S.is_client et P.type == Demande)
       ou (S.state ? OPEN et P.type == Demande et P.seqno ? S. OSR)
       ou (S.state ? OPEN et P.type == Réponse et P.seqno ? S. OSR)
       ou (S.state == RESPOND et P.type == Données),
      Envoyer des paquets Sync accusant réception de P.seqno
      Abandonner le paquet et retour

Étape 8 : Traitement des options et marquage comme acquittable
/* Le traitement des options n'est pas spécifiquement décrit ici. Certaines options, comme Obligatoire, peuvent provoquer
la réinitialisation de la connexion, auquel cas les étapes 9 et suivantes ne sont pas exécutées */
    Marquer les paquets comme acquittables (en termes de Vecteur Ack, reçus ou reçus marqués ECN)

Étape 9 : Traitement de Reset
    Si P.type == Reset,
      Supprimer la connexion
      S.state := TIMEWAIT
      Établir le temporisateur TIMEWAIT
      Abandonner le paquet et retour

Étape 10 : Traiter l'état DEMANDE (deuxième partie)
    Si S.state == DEMANDE,
/* Si on arrive ici, P est une réponse valable du serveur (voir Étape 4) et on devrait passer à l'état PARTOPEN. PARTOPEN
signifie envoyer un ACK, ne pas envoyer de paquets de données, retransmettre les Ack périodiquement, et toujours inclure
tout Init Cookie provenant de la réponse */
      S.state := PARTOPEN
      Établir le temporisateur PARTOPEN
      Continuer avec S.state == PARTOPEN
/* L'étape 12 enverra le Ack pour compléter la prise de contact en trois phases */

Étape 11 : Traiter l'état RÉPONDRE
    Si S.state == RESPOND,
      Si P.type == Demande,
        Envoyer Réponse, contenant éventuellement des Init Cookie
        Si un Init Cookie a été envoyé,
          Éliminer S et retour
/* L'étape 3 * va créer une autre prise lorsque le client achèvera la prise de contact à trois phases */
      Sinon,
        S.OSR := P.seqno
```

```
        S.state := OUVERT

Étape 12 : Traitement de l'état PARTOPEN
    Si S.state == PARTOPEN,
      Si P.type == Réponse,
        Envoyer ACK
      Sinon, si P.type != Sync,
        S. OSR := P.seqno
        S.state := OUVERT

Étape 13 : Traitement de CloseReq
    Si P.type == CloseReq et S.state < CLOSEREQ,
      Générer Fermer
      S.state: = CLOSING
      Établir le temporisateur CLOSING

Étape 14 : Traitement de Close
    Si P.type == Close,
      Générer Reset (Fermé)
      Supprimer la connexion
      Abandonner le paquet et retour

Étape 15 : Traitement de Sync
    Si P.type == Sync,
      Générer SyncAck

Étape 16 : Traiter les données
/* A ce stade, toute donnée d'application sur P peut être passée à l'application, sauf que l'application NE DOIT PAS recevoir
des données de plus d'une Demande ou Réponse */
```

9. Sommes de contrôle

DCCP utilise une somme de contrôle d'en-tête pour protéger son en-tête contre la corruption. Généralement, cette somme de contrôle couvre également toutes les données d'application. Les applications DCCP peuvent, toutefois, demander que la somme de contrôle d'en-tête couvre seulement une partie des données d'application, ou peut-être pas de données d'application du tout. Les couches de liaison peuvent alors réduire leur protection sur les parties non protégées des paquets DCCP. Pour certaines liaisons bruyantes, et pour les applications qui peuvent tolérer la corruption, ceci peut améliorer considérablement les taux de livraison et les performances perçues.

La couverture de somme de contrôle pourrait éventuellement impacter aussi les mécanismes de contrôle d'encombrement. Un paquet avec des données d'application corrompues et une couverture de somme de contrôle complète est traitée comme perdue. Cela fait subir une lourde charge de réponse de perte de la part du mécanisme de contrôle d'encombrement de l'envoyeur, ce qui peut pénaliser injustement les connexions sur les liens avec un contexte de corruption élevé. La combinaison des options de couverture de somme de contrôle réduite et de somme de contrôle des données peut laisser les points d'extrémité faire rapport des paquets comme corrompus plutôt qu'éliminés, en utilisant les options Données éliminées et le code d'abandon 3 (voir le paragraphe 11,7). Cela peut éventuellement bénéficier aux applications. Cependant, d'autres recherches sont nécessaires pour déterminer une réponse appropriée à la corruption, qui peut parfois se corréler avec l'encombrement. Les paquets corrompus reçoivent actuellement une réponse de perte.

L'option Somme de contrôle des données, qui contient un CRC fort, permet aux points d'extrémité de détecter la corruption des données d'application. Une API peut alors être utilisée pour éviter de livrer des données corrompues à l'application, même si les liaisons livrent des données corrompues au point d'extrémité en raison de la couverture réduite de somme de contrôle. Cependant, l'utilisation de la couverture réduite de somme de contrôle pour les applications qui demandent des données correctes est actuellement considérée comme expérimentale. Ceci parce que le taux de perte plus corruption combiné pour les paquets avec couverture réduite de somme de contrôle peut être significativement plus élevé que pour les paquets avec une couverture complète de somme de contrôle, bien que le taux de perte soit généralement inférieur. Le comportement réel dépendra de la conception des liaisons ; des recherches et une expérience supplémentaires sont requises.

La couverture réduite de somme de contrôle introduit quelques problèmes de sécurité; voir le paragraphe 18.1. Voir l'appendice B pour une discussion sur la motivation. La mise en œuvre DCCP de la couverture réduite de somme de contrôle a été inspirée par UDP-Lite [RFC3828].

9.1 Champ Somme de contrôle d'en-tête

DCCP utilise l'algorithme de somme de contrôle TCP/IP. Le champ Somme de contrôle dans l'en-tête DCCP générique (voir au paragraphe 5.1) est égal au complément à un de 16 bits de la somme des compléments à un de tous les mots de 16 bits dans l'en-tête DCCP, les options DCCP, un pseudo en-tête pris dans l'en-tête de la couche réseau, et, selon la valeur du champ Couverture de somme de contrôle, d'une partie ou de l'ensemble des données d'application. Lors du calcul de la somme de contrôle, le champ Somme de contrôle lui-même est traité comme 0. Si un paquet contient un nombre impair d'octets d'en-tête et de charge utile à additionner, 8 bits à zéro sont ajoutés sur la droite pour former un mot de 16 bits pour les besoins de la somme de contrôle. L'octet de bourrage n'est pas transmis au titre du paquet.

Le pseudo en-tête est calculé comme pour TCP. Pour IPv4, il est de 96 bits et se compose des adresses de source et destination IPv4, du numéro de protocole IP pour DCCP (bourré à gauche avec 8 bits à zéro) et la longueur DCCP comme une quantité de 16 bits (la longueur de l'en-tête DCCP avec les options, plus la longueur de toutes les données) ; voir la [RFC0793], paragraphe 3.1. Pour IPv6, il est de 320 bits, et se compose des adresses de source et destination IPv6, de la longueur DCCP comme une quantité de 32 bits et du numéro de protocole IP pour DCCP (bourré à gauche avec 24 bits à zéro) ; voir la [RFC2460], paragraphe 8.1.

Les paquets avec des sommes de contrôle d'en-tête invalides DOIVENT être ignorés. En particulier, leurs options NE DOIVENT PAS être traitées.

9.2 Champ Couverture de somme de contrôle d'en-tête

Le champ Couverture de somme de contrôle dans l'en-tête générique DCCP (voir paragraphe 5.1) spécifie quelles parties du paquet sont couvertes par le champ Somme de contrôle, comme suit :

CsCov = 0
Le champ Somme de contrôle couvre l'en-tête DCCP, les options DCCP, le pseudo en-tête de couche réseau, et toutes les données d'application dans le paquet, éventuellement bourrées à droite avec des zéros à un nombre pair d'octets.

CsCov = 1-15
Le champ Somme de contrôle couvre l'en-tête DCCP, les options DCCP, le pseudo en-tête de couche réseau, et les (CsCov-1) * 4 octets initiaux de données d'application du paquet.

Ainsi, si CsCov est 1, aucune des données de l'application n'est protégée par la somme de contrôle d'en-tête. La valeur (CsCov-1) * 4 DOIT être inférieure ou égale à la longueur des données d'application. Les paquets avec des valeurs CsCov invalides DOIVENT être ignorés, en particulier, leurs options NE DOIVENT PAS être traitées. La signification des valeurs autres que 0 et 1 doit être considérée comme expérimentale.

Les valeurs autres que 0 spécifient que la corruption est acceptable dans certaines ou l'ensemble des données d'application du paquet DCCP. En fait, DCCP ne peut même pas détecter la corruption dans les zones non couvertes par la somme de contrôle d'en-tête, à moins que l'option Somme de contrôle des données soit utilisée. Les applications ne devraient pas faire d'hypothèses concernant l'exactitude des données reçues qui ne sont pas couvertes par la somme de contrôle et devraient, si nécessaire, introduire leurs propres vérifications de validité.

Une interface d'application DCCP devrait laisser les applications envoyeuses suggérer une valeur pour CsCov pour les paquets envoyés, par défaut 0 (couverture complète). La caractéristique Minimum de couverture de somme de contrôle, décrite ci-dessous, permet à un point d'extrémité de refuser la livraison de données d'application sur les paquets avec une couverture partielle de somme de contrôle ; par défaut, seules les données d'application entièrement couvertes sont acceptées. Les couches inférieures qui prennent en charge la détection d'erreurs partielle PEUVENT utiliser le champ Couverture de somme de contrôle comme un indice de l'endroit où des erreurs n'ont pas besoin d'être détectées. Les couches inférieures DOIVENT utiliser un mécanisme de détection d'erreur fort permettant de détecter au moins les erreurs qui surviennent dans la partie sensible du paquet, et d'éliminer les paquets endommagés. La partie sensible se compose des octets entre le premier octet de l'en-tête IP et le dernier octet identifié par la couverture de somme de contrôle.

Pour plus de détails sur les questions d'interface d'application et de couche inférieure relatives à la somme de contrôle partielle, voir la [RFC3828].

9.2.1 Caractéristique Couverture minimum de somme de contrôle

La caractéristique Couverture minimum de somme de contrôle permet à un point d'extrémité DCCP de déterminer si son

homologue est prêt à accepter les paquets avec une couverture réduite de somme de contrôle. Par exemple, le DCCP A envoie une option "Change R (Couverture minimum de somme de contrôle, 1)" pour que DCCP B vérifie si B est prêt à accepter les paquets avec Couverture de somme de contrôle réglé à 1.

Couverture minimum de somme de contrôle a le numéro de caractéristique 8 et est à priorité du serveur. Il prend des valeurs entières d'un octet entre 0 et 15 ; les valeurs de 16 ou plus sont réservées. La couverture minimum de somme de contrôle/B reflète les valeurs de couverture de somme de contrôle que le DCCP B trouve inacceptables. Disons que la valeur de Couverture minimum de somme de contrôle/B soit MinCsCov. Alors :
- o Si MinCsCov = 0, DCCP B ne trouve que des paquets avec CsCov = 0 acceptables.
- o Si MinCsCov > 0, B DCCP trouve en plus des paquets avec CsCov ? MinCsCov acceptables.

DCCP B PEUT refuser de traiter les données d'application provenant de paquets avec une Couverture de somme de contrôle inacceptable. Ces paquets DEVRAIENT être signalés en utilisant les options Données abandonnées (paragraphe 11.7) avec le code d'abandon 0, Contraintes de protocole. Les nouvelles connexions commencent avec Couverture minimale de somme de contrôle 0 pour les deux points d'extrémité.

9.3 Option Somme de contrôle des données

L'option Somme de contrôle des données contient un code de vérification de redondance cyclique de 32 bits CRC-32c des données d'application d'un paquet DCCP.

Type = 44 ; Longueur = 6

Le DCCP qui envoie calcule le CRC des octets comprenant la zone des données d'application et le mémorise dans les données d'options. L'algorithme CRC-32c utilisé pour la somme de contrôle des données est le même que celui utilisé pour SCTP [RFC3309] ; noter que le CRC-32c de zéro octets de données est égal à zéro. La somme de contrôle d'en-tête DCCP couvrira l'option Somme de contrôle des données, de sorte que la somme de contrôle des données doit être calculée avant la somme de contrôle d'en-tête.

Un point d'extrémité DCCP qui reçoit un paquet avec une option Somme de contrôle des données DOIT ou PEUT vérifier la somme de contrôle des données ; le choix dépend de la valeur de la caractéristique Vérifier les données de somme de contrôle décrites ci-dessous. Si il vérifie la somme de contrôle, il calcule le CRC-32c des données d'application reçues à l'aide du même algorithme que l'envoyeur et compare le résultat à la valeur de la somme de contrôle des données. Si les CRC diffèrent, le point d'extrémité réagit d'une des deux façons suivantes :
- o L'application receveuse peut avoir demandé la livraison de données connues pour être corrompues par l'intermédiaire d'une API facultative. Dans ce cas, les données du paquet DOIVENT être transmises à l'application, en notant qu'elles sont connues pour être corrompues. Par ailleurs, le point d'extrémité de réception DOIT rapporter que le paquet est livré corrompu en utilisant une option Données abandonnées (Code d'abandon 7, "Livré corrompu").
- o Autrement, le point d'extrémité de réception DOIT abandonner les données d'application et rapporter les données comme éliminées pour cause de corruption en utilisant une option Données abandonnées (Code d'abandon 3, "Corrompu").

Dans les deux cas, le paquet est considéré comme acquittable (puisque son en-tête a été traité) et sera donc acquitté en utilisant l'équivalent des états Reçu ou Marqué reçu ECN de Vecteur Ack.

Bien que Somme de contrôle des données soit destiné aux paquets contenant des données d'application, il peut être inclus dans les autres paquets, comme DCCP-Ack, DCCP- Sync, et DCCP-SyncAck. Le receveur DEVRAIT calculer le CRC-32c de la zone des données d'application sur de tels paquets, tout comme il le fait pour les paquets DCCP-Données et similaires. Si les CRC diffèrent, les paquets DOIVENT de même être signalés en utilisant les options Données abandonnées (Code d'abandon 3) bien que leurs zones de données d'application ne soient jamais livrées à l'application.

9.3.1 Caractéristique Vérification de somme de contrôle des données

La caractéristique Vérification de somme de contrôle des données permet à un point d'extrémité DCCP de déterminer si son homologue va vérifier les options Somme de contrôle des données. Le DCCP A envoie une option obligatoire "Change R (Vérifier les données de contrôle, 1)" au DCCP B pour exiger qu'il vérifie les options Somme de contrôle des données (la connexion sera réinitialisation si il ne le peut pas).

Vérification des données de somme de contrôle a le numéro de caractéristique 9 et est une priorité de serveur. Elle prend

des valeurs booléennes sur un octet. Le DCCP B DOIT vérifier toutes les options Somme de contrôle de données reçues lorsque Vérifier la somme de contrôle des données/B est à un, bien qu'il PUISSE les vérifier, même si Vérifier la somme de contrôle des données/B est à zéro. Les valeurs de deux ou plus sont réservées. Les nouvelles connexions commencent avec Vérifier la somme de contrôle des données à 0 pour les deux points d'extrémité.

9.3.2 Notes sur l'utilisation de la somme de contrôle

Les liaisons Internet doivent normalement appliquer de fortes vérifications de contrôle d'intégrité aux paquets qu'elles transmettent [RFC3828], [RFC3819]. C'est le cas par défaut lorsque la valeur de la couverture de somme de contrôle de l'en-tête DCCP est égale à zéro (pleine couverture). Toutefois, la valeur de couverture de somme de contrôle DCCP pourrait ne pas être zéro. En établissant la couverture partielle de somme de contrôle, l'application indique qu'elle peut tolérer la corruption dans la partie non protégée des données d'application. Reconnaissant ce fait, les couches de liaison peuvent réduire la force de la détection d'erreur et/ou de correction lors de la transmission de cette partie non protégée. Ceci, à son tour, peut augmenter considérablement la probabilité que le point d'extrémité reçoive des données corrompues ; la somme de contrôle des données permet au receveur de détecter cette corruption avec une très forte probabilité .

10. Contrôle d’encombrement

Chaque mécanisme de contrôle d’encombrement pris en charge par DCCP se voit attribuer un identifiant de contrôle d’encombrement, ou CCID : un nombre de 0 à 255. Lors de l'établissement de la connexion et, éventuellement, par la suite, les points d'extrémité négocient leurs mécanismes de contrôle d’encombrement en négociant les valeurs de leurs caractéristiques d'identifiant de contrôle d’encombrement. L'identifiant de contrôle d’encombrement a un numéro de caractéristique de 1. La valeur du CCID/A est égale au CCID utilisé pour la demi-connexion A à B. Le DCCP B envoie une option "Change R(CCID, K)" pour demander au DCCP A d'utiliser le CCID K pour ses paquets de données.

Le CCID est une caractéristique de priorité de serveur, de sorte que les options de négociation de CCID peuvent comporter plusieurs CCID acceptables, triés par ordre décroissant de priorité. Par exemple, l'option "Change R(CCID, 2 3 4)" demande au receveur d'utiliser le CCID 2 pour ses paquets, bien que les CCID 3 et 4 soient également acceptables. (Ce qui correspond aux octets "35, 6, 1, 2, 3, 4" : Change R option (35), longueur d'option (6), ID de caractéristique (1), CCID (2, 3, 4)). De même, "Confirme L (CCID, 2, 2 3 4)" indique au receveur que l'envoyeur utilise le CCID 2 pour ses paquets, mais que les CCID 3 et 4 pourraient aussi être acceptables.

Les CCID actuellement affectés sont les suivants :

CCID	Signification	Référence
0-1	Réservé	
2	Contrôle d'encombrement de style TCP	[RFC4341]
3	Contrôle d'encombrement compatible avec TCP	[RFC4342]
4-255	Réservé	

Tableau 5 : Identifiants du contrôle d'encombrement DCCP

Les nouvelles connexions commencent par le CCID 2 pour les deux points d'extrémité. Si c'est inacceptable pour un point d'extrémité DCCP, celui-ci DOIT envoyer des options Changement obligatoire (CCID) sur ses premiers paquets.

Tous les CCID normalisés pour une utilisation avec DCCP correspondront à des mécanismes de contrôle d’encombrement précédemment normalisés par l'IETF. On s'attend à ce que pendant un certain temps, tous ces mécanismes soient compatibles avec TCP, mais la compatibilité avec TCP n'est pas une exigence explicite de DCCP.

Une mise en œuvre DCCP destinée à un usage général, comme une mise en œuvre dans un noyau polyvalent de système d'exploitation, DEVRAIT mettre en œuvre au moins CCID 2. L'intention est de faire que CCID 2 soit largement disponible pour l'interopérabilité, bien que des applications particulières puissent interdire son utilisation.

10.1 Contrôle d’encombrement de style TCP

Le CCID 2, contrôle d’encombrement de type TCP, note un contrôle d’encombrement à augmentation additive, un contrôle d’encombrement à diminution multiplicative (AIMD) avec le comportement modélisé directement sur TCP, incluant la fenêtre d’encombrement, le démarrage lent, les fins de temporisation, etc., [RFC2581]. Le CCID 2 atteint une bande passante maximale sur le long terme, compatible avec l'utilisation de bout en bout du contrôle d’encombrement, mais diminue de moitié sa fenêtre d’encombrement en réponse à chaque événement d’encombrement. Cela conduit aux

changements brusques de taux typiques de TCP. Les applications devraient utiliser le CCID 2 si elles préfèrent l'utilisation d'une bande passante maximale à la constance du taux. C'est souvent le cas pour les applications qui n'exécutent pas leurs données directement à l'utilisateur.

Par exemple, une application hypothétique qui a transféré des fichiers sur DCCP, en utilisant des retransmissions au niveau application pour les paquets perdus, préfèrerait CCID 2 à CCID 3. Les jeux en ligne peuvent également préférer le CCID 2.

Le CCID 2 est décrit plus en détails dans la [RFC4341].

10.2. Contrôle d'encombrement TFRC

CCID 3 désigne le contrôle d'encombrement compatible avec TCP (TFRC, *TCP-Friendly Rate Control)* un mécanisme de contrôle d'encombrement à débit contrôlé fondé sur une équation. TFRC est conçu pour être raisonnablement équitable lorsque il y a concurrence pour la bande passante avec des flux de type TCP, où un flux est "raisonnablement équitable" si son taux d'envoi est généralement dans un facteur deux du taux d'envoi d'un flux TCP sous les mêmes conditions. Toutefois, le TFRC a une variation de débit beaucoup plus faible au cours du temps par rapport à TCP, ce qui rend le CCID 3 plus adapté que CCID 2 pour des applications telles que des supports en flux continus où un taux d'envoi relativement lisse est important.

Le CCID 3 est décrit plus en détails dans la [RFC4342]. Les algorithmes de contrôle d'encombrement TFRC ont été initialement décrits dans la [RFC3448].

10.3 -Options, caractéristiques et codes de réinitialisation spécifiques de CCID

La moitié des types d'options, des numéros de caractéristique et des codes de réinitialisation sont réservés pour des utilisations spécifiques de CCID. Les CCID peuvent souvent avoir besoin de nouvelles options, pour communiquer des informations de débit ou d'accusé de réception, par exemple ; les espaces d'option réservés permettent aux CCID de créer des options à volonté sans polluer l'espace global d'options. L'option 128 pourrait avoir différentes significations sur une demi-connexion utilisant le CCID 4 et une demi-connexion utilisant le CCID 8. Les options et caractéristiques de CCID spécifiques ne seront jamais en conflit avec des options globales et des caractéristiques introduites par les versions ultérieures de la présente spécification.

Tout paquet peut contenir des informations destinées à l'une ou l'autre demi-connexion, de sorte que les types d'options spécifiques de CCID, les numéros de caractéristique, et les codes de réinitialisation signalent explicitement la demi-connexion à laquelle ils s'appliquent.

- Les numéros d'option 128 à 191 sont pour les options envoyées par la demi-connexion envoyeur à la demi-connexion receveur ; les numéros d'option 192 à 255 sont pour les options envoyées par la demi-connexion receveur à la demi-connexion envoyeur.

- Les codes de réinitialisation 128 à 191 indiquent que la demi-connexion envoyeur réinitialise la connexion (probablement à cause de certains problèmes avec les accusés de réception envoyés par la demi-connexion receveur). Les codes de réinitialisation 192 à 255 indiquent que la demi-connexion receveur réinitialise la connexion (probablement à cause de certains problèmes avec des paquets de données envoyés par la demi-connexion envoyeur).

- Enfin, les numéros de caractéristique de 128 à 191 sont utilisés pour les caractéristiques situées à la demi-connexion envoyeur ; les numéros de caractéristique de 192 à 255 sont pour les caractéristiques situées à la demi-connexion receveur. Comme les options Change L et Confirme L pour une caractéristique sont envoyées par la localisation des caractéristiques, on sait que toute option Change L(128) a été envoyée par la demi-connexion envoyeur, alors que toute option Change L(192) a été envoyée par la demi-connexion receveur. De même, les options Change R(128) sont envoyées par la demi-connexion receveur, tandis que les options Change R(192) sont envoyées par la demi-connexion envoyeur.

Par exemple, considérons une connexion DCCP où la demi-connexion A à B utilise le CCID 4 et la demi-connexion B à A utilise le CCID 5. Voici comment un échantillonnage d'options spécifiques de CCID est alloué aux demi-connexions.

Paquet	Option	demi connexion pertinente	CCID pertinent
A > B	128	A vers B	4
A > B	192	B vers A	5
A > B	Change L(128, ...)	A vers B	4
A > B	Change R(192, ...)	A vers B	4

A > B	Confirme L(128, ...)	A vers B	4
A > B	Confirme R(192, ...)	A vers B	4
A > B	Change R(128, ...)	B vers A	5
A > B	Change L(192, ...)	B vers A	5
A > B	Confirme R(128, ...)	B vers A	5
A > B	Confirme L(192, ...)	B vers A	5
B > A	128	B vers A	5
B > A	192	A vers B	4
B > A	Change L(128, ...)	B vers A	5
B > A	Change R(192, ...)	B vers A	5
B > A	Confirme L(128, ...)	B vers A	5
B > A	Confirme R(192, ...)	B vers A	5
B > A	Change R(128, ...)	A vers B	4
B > A	Change L(192, ...)	A vers B	4
B > A	Confirme R(128, ...)	A vers B	4
B > A	Confirme L(192, ...)	A vers B	4

Utiliser des options spécifiques de CCID et des options de caractéristique lors d'une négociation pour la caractéristique de CCID correspondante N'EST PAS RECOMMANDÉ, car il est difficile de prédire quel CCID sera en vigueur lorsque l'option sera traitée. Par exemple, si une Demande-DCCP contient la séquence d'options "Change L(CCID, 3), 128", l'option spécifique de CCID "128" peut être traitée soit par le CCID 3 (si le serveur prend en charge le CCID 3) soit par la valeur par défaut de CCID 2 (si il ne le fait pas). Cependant, il est sûr d'inclure des options spécifiques de CCID à la suite de certaines options de changement obligatoire (CCID). Par exemple, si une Demande-DCCP contient la séquence d'options "Obligatoire, Change L(CCID, 3), 128", alors soit l'option "128" sera traitée par le CCID 3, soit la connexion sera réinitialisée.

Les serveurs qui ne mettent pas en œuvre l'option par défaut CCID 2 pourraient néanmoins recevoir des options spécifiques de CCID 2 sur un paquet Demande-DCCP. (Un tel serveur DOIT envoyer des options Changement obligatoire (CCID) sur ses DCCP-Réponse, afin que les options spécifiques de CCID sur tout autre paquet ne se réfèrent pas à CCID 2.) Le serveur DOIT traiter de telles options comme non comprises. Ainsi, il va réinitialiser la connexion lorsque il rencontre une option obligatoire spécifique de CCID ou une demande de négociation de caractéristiques, envoyer un Confirme vide pour une option Changement non obligatoire pour une caractéristique spécifique de CCID, et ignorer les autres options spécifiques de CCID.

10.4 Exigences de profil de CCID

Chaque document de profil de CCID DOIT traiter au moins les exigences suivantes :

- Le profil doit inclure le nom et le numéro du CCID décrit.
- Le profil DOIT décrire les conditions dans lesquelles il est susceptible d'être utilisé. Souvent, la meilleure façon de le faire est par rapport aux CCID existants.
- Le profil DOIT énumérer et décrire toutes les options spécifiques de CCID, les caractéristiques, et les codes de réinitialisation et DEVRAIT énumérer les options générales et les caractéristiques décrites dans ce document qui sont particulièrement pertinentes pour le CCID.
- Tout mécanisme d'acquittement nouvellement défini DOIT inclure un moyen de retransmettre des échos de nom occasionnel ECN à l'envoyeur.
- Le profil DOIT décrire le format de paquets de données, y compris les options qui devraient être incluses et le réglage du champ d'en-tête CCval.
- Le profil DOIT décrire le format des paquets d'accusé de réception, y compris toutes les options qui devraient être incluses.
- Le profil DOIT définir comment les paquets de données sont contrôlés par rapport à l'encombrement. Cela inclut les réponses aux événements d'encombrement, aux périodes d'inactivité et d'application limitée, et aux options DCCP Données abandonnées et Réception lente. Les CCID qui mettent en œuvre le contrôle d'encombrement par paquet DEVRAIENT préciser la façon dont la taille de paquet est prise en compte dans les décisions de contrôle d'encombrement.
- Le profil DOIT spécifier quand les paquets d'accusé de réception sont générés et comment ils sont contrôlés quant à l'encombrement.
- Le profil DOIT définir quand un envoyeur utilisant le CCID est considéré comme passant au repos.
- Le profil DOIT dire si les accusés de réception de son CCID ont besoin de d'être acquittés et, si oui, à quelle fréquence.

10.5 État d’encombrement

La plupart des algorithmes de contrôle d’encombrement dépendent de l'histoire passée pour déterminer le taux d'envoi autorisé actuel. Dans CCID 2, cet état d’encombrement comprend une fenêtre d’encombrement et une mesure du nombre de paquets en transit dans le réseau ; dans CCID 3, il inclut les longueurs des intervalles de pertes récentes. Les deux CCID utilisent une estimation du temps d'aller-retour. L'état d’encombrement dépend du chemin du réseau et est invalidé par les changements de chemin. Par conséquent, les envoyeurs et receveurs DCCP DEVRAIENT réinitialiser leur état d’encombrement -- essentiellement en redémarrant le contrôle d’encombrement à partir du "démarrage lent" ou équivalent -- lors des changements significatifs dans le chemin de bout en bout. Par exemple, un point d'extrémité qui envoie ou reçoit un message Mise à jour de lien IPv6 mobile [RFC3775] DEVRAIT réinitialiser son état d’encombrement pour toutes les connexions DCCP correspondantes.

Une mise en œuvre DCCP PEUT également réinitialiser son état d’encombrement quand un CCID change (c'est-à-dire, quand une négociation pour une caractéristique de CCID s'achève avec succès et que la nouvelle valeur de caractéristique diffère de l'ancienne valeur). Donc, une connexion dans un environnement fortement encombré pourrait échapper au contrôle d’encombrement de bout en bout par une renégociation fréquente de CCID, tout comme elle pourrait échapper au contrôle d’encombrement de bout en bout par l'ouverture de nouvelles connexions pour la même session. Ce comportement est interdit. Pour l'empêcher, les mises en œuvre DCCP PEUVENT limiter le taux auquel le CCID peut être changé -- par exemple, en refusant de modifier une valeur de caractéristique de CCID plus d'une fois par minute.

11. Accusés de réception

Le contrôle d’encombrement exige que les receveurs transmettent des informations sur les pertes de paquets et les marques ECN aux envoyeurs. Les receveurs DCCP DOIVENT signaler tous les encombrements qu'ils voient, comme défini par le profil CCID pertinent. Chaque CCID dit quand les accusés de réception devraient être envoyés, quelles sont les options qu'ils doivent utiliser, et ainsi de suite. Les accusés de réception DCCP sont contrôlés quant à l’encombrement, mais il n'est pas nécessaire que le flux d’accusés de réception soit plus que très grossièrement conforme à TCP ; chaque CCID définit comment les accusés de réception sont contrôlés quant à l’encombrement.

La plupart des accusés de réception utilisent les options DCCP. Par exemple, sur une demi-connexion avec CCID 2 (style TCP) le receveur fait rapport des informations d'accusé de réception en utilisant l'option Vecteur Ack. Cette Section décrit les options communes d'accusé de réception et montre comment les accusés de réception qui utilisent ces options fonctionnent habituellement. La description complète des mécanismes d'accusé de réception utilisés pour chaque CCID est énoncée dans les spécifications de profil CCID.

Les options d'accusé de réception, telles que Vecteur Ack, dépendent du numéro d'accusé de réception DCCP et ne sont donc autorisées que sur les types de paquets qui portent ce numéro. Les options d'accusé de réception reçues sur d'autres types de paquets, à savoir Demande-DCCP et DCCP-Données, DOIVENT être ignorées. Des options d'accusé de réception détaillées ne sont cependant pas nécessairement exigées sur chaque paquet qui porte un numéro d'accusé de réception.

11.1 Accusés de réception des accusés de réception et connexions unidirectionnelles

DCCP a été conçu pour bien fonctionner sur les flux de données bidirectionnels et unidirectionnels, et pour les connexions qui transitent entre ces états. Toutefois, les accusés de réception requis pour une connexion unidirectionnelle sont très différents de ceux requis pour une connexion bidirectionnelle. En particulier, les connexions unidirectionnelles ont besoin de se soucier des accusés de réception d'accusés de réception (acc d'acc).

Le problème des acc d'acc se pose parce que certains mécanismes d'accusé de réception sont fiables. Par exemple, une demi-connexion receveur utilisant le CCID 2, contrôle d'encombrement de type TCP, envoie des Vecteurs Ack contenant des informations d'accusé de réception complètement fiables. La demi-connexion envoyeur devrait occasionnellement informer la demi-connexion receveur qu'elle a reçu un ACK. Si elle ne le fait pas, la demi-connexion receveur pourrait renvoyer les informations complètes de Vecteur Ack, remontant au début de la connexion, avec tous les paquets DCCP-Ack ! Cependant, on notera que les acc d'acc n'ont pas besoin d'être fiables eux-mêmes : quand un acc d'acc est perdu, la demi-connexion receveur conserve simplement, et retransmet périodiquement, le vieil état lié aux accusés de réception pour un peu plus longtemps. Par conséquent, il n'est pas besoin d'acc d'acc d'acc.

Lorsque la communication est bidirectionnelle, tous les acc d'acc nécessaires sont automatiquement contenus dans les accusés de réception normaux pour les paquets de données. Sur une connexion unidirectionnelle cependant, le receveur DCCP n'envoie pas de données, de sorte que l'envoyeur ne va normalement pas envoyer d'accusé de réception. Par conséquent, le CCID en vigueur sur cette demi-connexion doit explicitement dire si, quand, et comment la demi-connexion envoyeur devrait générer des acc d'acc.

Par exemple, considérons une connexion bidirectionnelle où les deux demies connexions utilisent le même CCID (2 ou 3) et où le DCCP B passe à "Repos". Cela signifie que la connexion devient unidirectionnelle :

DCCP B arrête l'envoi de données et envoie seulement des paquets DCCP-Ack à DCCP A. En CCID 2, contrôle d'encombrement de type TCP, DCCP B utilise le Vecteur d'Ack pour communiquer de manière fiable quels paquets il a reçus. Comme décrit ci-dessus, DCCP A doit parfois accuser réception d'un pur accusé de réception provenant de DCCP B de sorte que B puisse libérer le vieil état de Vecteur Ack. Par exemple, A peut envoyer un paquet DCCP-DataAck au lieu d'un DCCP-Données de temps en temps. Dans CCID 3, cependant, l'état d'accusé de réception est généralement borné, de sorte que A n'a pas besoin d'accuser réception des accusés de réception de B.

Lorsque la communication est unidirectionnelle, un seul CCID -- dans l'exemple, le CCID de A à B -- contrôle les accusés de réception des deux DCCP, en termes de leur contenu, leur fréquence, et ainsi de suite. Pour les connexions bidirectionnelles, le CCID de A à B régit les accusés de réception de DCCP B (y compris ses accusés de réception des accusés de réception du DCCP A) et le CCID de B à A régit les accusés de réception du DCCP A.

DCCP A passe son modèle d'accusé de réception de bidirectionnel à unidirectionnel quand il remarque que DCCP B est passé au repos. Il passe de unidirectionnel à bidirectionnel quand il doit accuser réception même d'un seul paquet DCCP-Données ou DCCP-DataAck provenant de DCCP B.

Chaque CCID définit la façon de détecter le repos sur ce CCID, et comment ce CCID traite les acc d'acc sur les connexions unidirectionnelles. Le CCID de B à A définit quand le DCCP B est passé au repos. Habituellement, cela se produit quand une certain délai est passé sans que B envoie de paquet de données ; en CCID 2, par exemple, cette période est le maximum de 0,2 seconde et deux temps d'aller-retour. Le CCID de A à B définit comment le DCCP A traite les acc d'acc une fois que le DCCP B est passé au repos.

11.2 Portage des accusés de réception

Les accusés de réception de données de A à B PEUVENT être portés sur les données envoyées par DCCP B, pour autant que cela ne retarde pas l'accusé de réception plus que ce que le CCID de A à B trouvera acceptable. Cependant, les accusés de réception de données ont souvent besoin de plus de 4 octets pour s'exprimer. Un grand ensemble d'accusés de réception ajouté à un paquet de données important pourrait dépasser la taille maximale autorisée des paquets. Dans ce cas, DCCP B DEVRAIT envoyer séparément les paquets DCCP-Données et les paquets DCCP-Ack, ou attendre, mais pas trop longtemps, un datagramme plus petit.

Le portage est particulièrement fréquent au DCCP A quand la demi connexion B vers A est au repos – c'est-à-dire, quand le DCCP A accuse simplement réception des accusés de réception du DCCP B. Il y a trois raisons pour accuser réception des accusés de réception du DCCP B : pour permettre au DCCP B de libérer les informations sur des paquets de données précédemment acquittés provenant de A ; pour réduire la taille des accusés de réception à venir, et pour manipuler le taux d'envoi des accusés de réception à venir. Comme ce sont des préoccupations secondaires, le DCCP A peut généralement se permettre d'attendre indéfiniment un paquet de données pour porter ses accusés de réception ; si DCCP B veut susciter un accusé de réception, il peut envoyer un DCCP-Sync.

Toute les restrictions au portage d'accusé de réception sont décrites dans le profil du CCID pertinent.

11.3. Caractéristique Taux d'accusés de réception

La caractéristique Taux d'accusés de réception *(Ack Ratio)* permet aux demi-connexions envoyeur d'influencer le taux auquel les demi-connexions receveur génèrent les paquets DCCP-Ack, contrôlant ainsi l'encombrement sur le chemin inverse. Cela diffère de TCP, qui n'a actuellement pas de contrôle d'encombrement pour le trafic pur d'accusé de réception. Le contrôle d'encombrement du taux d'accusés de réception sur le chemin inverse ne cherche pas à être conforme à TCP. Il essaie juste de éviter l'embouteillage complet, et de faire un peu mieux que TCP en présence d'un fort taux de perte ou de marques de paquets sur le chemin inverse.

Ratio Ack s'applique aux CCID dont l' horloge des demi-connexions receveuses accuse réception des paquets de données. La valeur du ratio Ack/A est égal au ratio approximatif de paquets de données envoyés par le DCCP A aux paquets DCCP-Ack envoyés par le DCCP B. Des ratios plus élevés correspondent à des taux de DCCP-Ack inférieurs ; l'envoyeur relève le Ratio Ack lorsque le chemin inverse est congestionné et abaisse le Ratio Ack quand il ne l'est pas. Chaque profil CCID définit comment il contrôle l'encombrement sur le chemin de l'accusé de réception, et en particulier, quel que soit le Ratio Ack utilisé. Le CCID 2, par exemple, utilise Ratio Ack pour le contrôle d'encombrement d'accusés de réception, mais CCID 3 ne le fait pas. Toutefois, chaque caractéristique de Ratio Ack a une valeur, que cette valeur soit utilisée ou non par

le CCID pertinent.

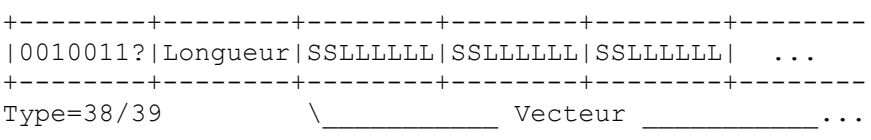
Ratio Ack a le numéro de caractéristique 5 et est non négociable. Il prend des valeurs entières de deux octets. Une valeur de Ratio Ack/A de quatre signifie que le DCCP B enverra au moins un paquet d'accusé de réception pour tous les quatre paquets de données envoyés par le DCCP A. Le DCCP A envoie une option "Change L(Ratio Ack)" pour notifier au DCCP B son ratio d'accusés de réception. Une valeur de Ratio Ack de zéro indique que la demi-connexion pertinente n'utilise pas de Ratio Ack pour contrôler son taux d'accusé de réception. Les nouvelles connexions commencent par un Ack Ratio de 2 pour les deux points d'extrémité ; ce Ratio Ack résulte en un comportement d'accusé de réception analogue aux accusés de réception retardés de TCP.

Un Ratio Ack devrait être traité comme une ligne directrice plutôt que comme une stricte exigence. Il est prévu que le comportement d'accusé de réception contrôlé par Ack Ratio ressemble à un comportement d'accusé de réception de TCP quand il n'y a pas d'encombrement sur le chemin inverse, et soit un peu plus prudent quand il y a de l'encombrement sur le chemin inverse. Suivre cette intention est plus important que la mise en œuvre précise de Ratio Ack. En particulier :

- Les receveurs PEUVENT porter les informations d'accusé de réception sur les paquets de données, en créant des paquets DCCP-DataAck. Le Ratio Ack ne s'applique pas aux accusés de réception portés. Toutefois, si les paquets de données sont trop gros pour transporter des informations d'accusé de réception, ou si le taux d'envoi des données est inférieur à ce que Ratio Ack suggère, le DCCP B devrait alors envoyer assez de purs paquets DCCP-Ack pour maintenir le taux d'un accusé de réception par Ratio Ack de paquet de données reçu.
- Les receveurs PEUVENT ralentir le taux de leurs accusés de réception plutôt que d'envoyer les accusés de réception immédiatement après la réception de paquets de données. Les receveurs qui ralentissent ce taux d'accusés de réception DEVRAIENT choisir un taux qui se rapproche de l'effet du Ratio Ack et DEVRAIENT inclure des options Temps écoulé (paragraphe 13.2) pour aider l'envoyeur à calculer le temps d'aller-retour.
- Les receveurs DEVRAIENT mettre en œuvre des temporisateurs de retard d'accusé de réception comme ceux de TCP, par lesquels l'accusé de réception de tout paquet est retardé d'au plus T secondes. Ce délai permet au receveur de recueillir des paquets supplémentaires à acquitter et de réduire ainsi les frais généraux par paquet des accusés de réception ; mais si T secondes ont passé et si l'accusé de réception est encore là, il est envoyé immédiatement. La valeur par défaut de T devrait être de 0,2 seconde, comme cela est courant dans les mises en œuvre de TCP. Cela peut conduire à l'envoi de plus de paquets d'accusé de réception que ce que le Ratio Ack pourrait suggérer.
- Les receveurs DEVRAIENT envoyer immédiatement les accusés de réception à réception des paquets marqués Encombrement ECN rencontré ou des paquets dont le numéro de séquence déclassé indique potentiellement une perte. Toutefois, il n'est pas nécessaire d'envoyer de tels accusés de réception immédiats pour les paquets marqués plus d'une fois par temps d'aller-retour.
- Les receveurs PEUVENT ignorer le Ratio Ack si ils effectuent leur propre contrôle d'encombrement sur les accusés de réception. Par exemple, un receveur qui connaît le taux de perte et de marque pour ses paquets DCCP-Ack pourraient maintenir de lui-même un taux d'acquittement compatible TCP. Un tel receveur DOIT s'assurer qu'il obtient toujours suffisamment d'informations de perte d'accusés de réception et de marques ou revenir au Ratio Ack lorsque des informations suffisantes ne sont pas disponibles, comme cela pourrait se produire pendant les périodes où le receveur est au repos.

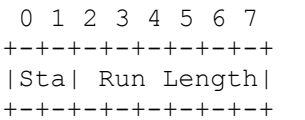
11.4 Options de vecteur d'accusé de réception

Le vecteur d'accusé de réception donne un historique codé des paquets de données reçus chez le client. Chaque octet du vecteur donne l'état de ce paquet de données dans l'historique des pertes, et le nombre de paquets précédents avec le même état. Les données de l'option ressemblent à ceci :



Les deux options Vecteur Ack (types d'options 38 et 39) ne diffèrent que par les valeurs qu'elles impliquent pour le nom occasionnel d'écho ECN *(ECN Echo Nonce)*. Le paragraphe 12.2 décrit cette question plus en détails.

Le vecteur lui-même se compose d'une série d'octets, dont le codage est le suivant :



Sta(TE) occupe les deux bits de poids fort de chaque octet et peut avoir l'une des quatre valeurs suivantes :

État	Signification
0	Reçu
1	Reçu avec marquage ECN
2	Réservé
3	Pas encore reçu

Tableau 6 : États du vecteur d'accusé de réception DCCP

Le terme "Reçu avec marquage ECN" se réfère à des paquets avec le codet ECN 11, CE (Encombrement rencontré) ; les paquets reçus avec ce codet ECN DOIVENT être signalés en utilisant l'état 1, Reçu avec marquage ECN. Les paquets reçus avec les codets ECN 00, 01 ou 10 (respectivement Non-ECT, ECT (0), ou ECT (1)) doivent être rapportés en utilisant l'état 0, Reçu.

Run Length *(longueur de plage)*, les six bits de moindre poids de chaque octet, spécifient combien de paquets consécutifs ont l'état donné. Run Length zéro dit que l'état correspondant s'applique à un paquet seulement ; Run Length 63 dit qu'il s'applique à 64 paquets consécutifs. Run length de 65 ou plus doit être codé sur plusieurs octets.

Le premier octet de la première option Vecteur Ack se réfère au paquet indiqué dans le numéro d'accusé de réception ; les octets ultérieurs font référence aux paquets plus anciens. Vecteur Ack NE DOIT PAS être envoyé sur les paquets DCCP-Données et Demande-DCCP, qui n'ont pas de numéro d'accusé de réception, et toutes les options Vecteur Ack rencontrées sur de tels paquets doivent être ignorées.

Un Vecteur Ack contenant les valeurs décimales 0,192,3,64,5 et pour lequel le numéro d'accusé de réception est le décimal 100 indique que :
- le paquet 100 a été reçu (numéro d'accusé de réception 100, état 0, Run Length 0) ;
- le paquet 99 a été perdu (état 3, Run Length 0) ;
- les paquets 98, 97, 96 et 95 ont été reçus (état 0, Run Length 3) ;
- le paquet 94 a été marqué ECN (état 1, Run Length 0) ;
- les paquets 93, 92, 91, 90, 89, et 88 ont été reçus (état 0, Run Length 5).

Une seule option Vecteur Ack peut accuser réception de jusqu'à 16 192 paquets de données. Si plus de paquets ont besoin d'être acquittés que ce qui peut tenir dans 253 octets de Vecteur Ack, plusieurs options Vecteur Ack peuvent être envoyées ; le second Vecteur Ack commence là où le premier s'est arrêté, et ainsi de suite.

Les états de Vecteur Ack sont soumis à deux contraintes générales. (Ces principes DEVRAIENT également être suivis pour les autres mécanismes d'accusé de réception ; se référer aux états de Vecteur Ack simplifie leur explication.)

1. Les paquets signalés comme état 0 ou état 1 DOIVENT être acquittables : leurs options ont été traitées par la pile DCCP receveuse. Il n'est pas nécessaire que toutes les données sur le paquet aient été livrées à l'application receveuse ; en fait, les données peuvent avoir été abandonnées.

2. Les paquets signalés comme état 3 NE DOIVENT PAS être acquittables. Les négociations de caractéristiques et les options sur de tels paquets NE DOIVENT PAS avoir été traitées, et le numéro d'accusé de réception NE DOIT PAS correspondre à un tel paquet.

Les paquets abandonnés dans une mémoire de réception de l'application DOIVENT être rapportés comme reçus ou reçus marqués ECN (états 0 et 1) selon leur état ECN ; les noms occasionnels ECN de tels paquets DOIT être inclus dans l'écho de nom occasionnel *(Nonce Echo)*. L'option Données abandonnées informe l'envoyeur que certains paquets signalés comme reçus ont en fait eu leurs données d'application abandonnées.

Une ou plusieurs options Vecteur Ack qui, ensemble, rapportent l'état d'un paquet avec un numéro de séquence inférieur à l'ISN, le numéro de séquence initial, DEVRAIENT être considérées comme invalides. Le DCCP receveur DEVRAIT soit ignorer les options, soit réinitialiser la connexion avec le code de réinitialisation 5, "Erreur d'option". Aucune option Vecteur Ack ne peut se référer à un paquet qui a pas encore été envoyé, comme l'assurent les contrôles de numéro d'accusé de réception du paragraphe 7.5.3, mais à cause d'une attaque, d'une erreur de mise en œuvre, ou d'un mauvais comportement, une option Vecteur Ack peut prétendre qu'un paquet a été reçu avant qu'il ne soit effectivement livré. Le paragraphe 12.2 décrit comment ceci est détecté et comment les envoyeurs devraient réagir. Les paquets qui n'ont pas été inclus dans une option Vecteur Ack DEVRAIENT être traités comme "pas encore reçu"(état 3) par l'envoyeur.

L'Annexe A fournit une description non normative des détails du traitement de l'accusé de réception DCCP dans le contexte d'une mise en œuvre abstraite de Vecteur Ack.

11.4.1 Cohérence du vecteur d'accusé de réception

Un émetteur va généralement recevoir des accusés de réception DCCP multiples pour certains de ses paquets de données. Par exemple, une demi-connexion envoyeur pourrait recevoir deux DCCP-Ack avec des Vecteur Ack, contenant tous deux des informations sur le numéro de séquence 24. (Les informations sur un numéro de séquence sont généralement répétées dans chaque accusé de réception jusqu'à ce que la demi-connexion envoyeur accuse réception d'un Ack. Dans ce cas, peut-être la demi-connexion receveur envoie les Ack plus rapidement que la demi-connexion envoyeur HC n'en accuse réception.) Dans un monde parfait, les deux Vecteurs Ack seraient toujours cohérents. Cependant, il y a beaucoup de raisons pour lesquelles ils ne pourraient pas l'être. Par exemple :

- La demi-connexion a reçu le paquet 24 entre l'envoi de ses accusés de réception, de sorte le premier Ack dit que 24 n'a pas été reçu (état 3) et le second dit qu'il a été reçu ou marqué ECN (état 0 ou 1).
- La demi-connexion receveur a reçu le paquet 24 entre l'envoi de ses accusés de réception, et le réseau a réorganisé les accusés de réception. Dans ce cas, le paquet va apparaître comme transitant de l'état 0 ou 1 à l'état 3.
- Le réseau a dupliqué le paquet 24, et l'un des doublons a été marqué ECN. Cela pourrait apparaître comme une transition entre les états 0 et 1.

Pour faire face à ces situations, les mises en œuvre de demi-connexion DCCP envoyeur DEVRAIENT combiner plusieurs états de Vecteur Ack reçus selon le tableau suivant :

```
          État reçu
          0   1   3
        +---+---+---+
      0 | 0 |0/1| 0 |
Vieil   +---+---+---+
      1 | 1 | 1 | 1 |
état    +---+---+---+
      3 | 0 | 1 | 3 |
        +---+---+---+
```

Pour lire le tableau, choisir la ligne correspondant à l'ancien état du paquet et la colonne correspondant à l'état du paquet dans le nouveau Vecteur Ack reçu, puis lire le nouvel état du paquet sur le tableau. Pour un ancien état de 0 (reçus non marqué) et l'état reçu de 1 (reçu marqué ECN) le nouvel état du paquet peut être réglé soit à 0, soit à 1. La mise en œuvre de demi-connexion envoyeur sera indifférente à la réorganisation si elle choisit un nouvel état pour cette cellule.

La demi-connexion receveur devrait collecter les informations sur les paquets reçus selon le tableau suivant :

```
            Paquet reçu
             0   1   3
           +---+---+---+
         0 | 0 |0/1| 0 |
   État    +---+---+---+
         1 |0/1| 1 | 1 |
mémorisé   +---+---+---+
         3 | 0 | 1 | 3 |
           +---+---+---+
```

Ce tableau est égal au tableau d'envoyeur, sauf que, lorsque l'état mémorisé est 1 et que l'état reçu est 0, le receveur est autorisé à basculer son état mémorisé à 0.

Une demi-connexion envoyeur PEUT choisir d'éliminer les anciennes informations glanées des Vecteurs Ack de la demi-connexion receveur, auquel cas elle DOIT ignorer les accusés de réception nouvellement reçus de la demi-connexion receveur pour ces vieux paquets. Il est souvent plus sage de sauvegarder les dernières informations de Vecteur Ack pendant un moment de sorte que la demi-connexion envoyeur puisse annuler sa réaction à un encombrement présumé quand un paquet "perdu" réapparaît de manière inattendue (transition de l'état 3 à l'état 0).

11.4.2 Couverture du vecteur d'accusé de réception

On peut diviser les paquets qui ont été envoyés à partir d'une demi-connexion envoyeur à une demi-connexion receveur en quatre groupes à peu près contigus. Du plus ancien au plus récent, ce sont :

1. Les paquets déjà acquittés par la demi-connexion receveur, où celle ci sait que la demi-connexion envoyeur a certainement reçu les accusés de réception ;
2. Les paquets déjà acquittés par la demi-connexion receveur, où celle-ci ne peut pas être sûre que la demi-connexion envoyeur a reçu les accusés de réception ;

3. Les paquets non encore acquittés par la demi-connexion receveur ;
4. Les paquets non encore reçus par la demi-connexion receveur.

L'union des groupes 2 et 3 est appelée la fenêtre d'accusé de réception. Généralement, chaque Vecteur Ack généré par la demi-connexion receveur couvrira toute la fenêtre d'accusé de réception : les accusés de réception de Vecteur Ack sont cumulatifs. (Cela simplifie la maintenance de Vecteur Ack à la demi-connexion receveur ; voir l'Appendice A). Lorsque les paquets sont reçus, cette fenêtre s'étend sur la droite et se rétrécit sur la gauche. Elle s'étend parce qu'il y a plus de paquets, et se rétrécit parce que les numéros d'accusé de réception de la demi-connexion envoyeur accusent réception des accusés de réception antérieurs, déplaçant les paquets du groupe 2 dans le groupe 1.

11.5 Caractéristique Envoyer Vecteur d'accusé de réception

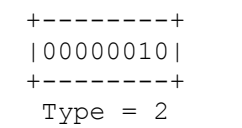
La caractéristique Envoyer Vecteur Ack permet aux DCCP de négocier si elles devraient utiliser les options Vecteur Ack pour signaler l'encombrement. Vecteur Ack fournit des informations détaillées sur les pertes et permet aux envoyeurs de rapporter à leurs applications si des paquets particuliers ont été abandonnés. Envoyer Vecteur Ack est obligatoire pour certains CCID et facultative pour les autres.

Envoyer Vecteur Ack a le numéro de caractéristique 6 et est une priorité de serveur. Il prend des valeurs booléennes sur un octet. Le DCCP A DOIT envoyer des options Vecteur Ack sur ses accusés de réception lorsque Envoyer Vecteur Ack/A a la valeur 1, bien qu'il PUISSE envoyer des options Vecteur Ack même lorsque Envoyer Vecteur Ack/A est 0. Les valeurs de deux ou plus sont réservées. Les nouvelles connexions commencent par Envoyer Vecteur Ack à 0 pour les deux points d'extrémité. Le DCCP B envoie une option "Change R(Envoyer Vecteur Ack, 1)" au DCCP A pour demander à A d'envoyer des options Vecteur Ack au titre de son trafic d'accusés de réception.

11.6 Option Receveur lent

Une demi-connexion receveur envoie l'option Receveur lent à son envoyeur pour indiquer qu'il a du mal à suivre les données de l'envoyeur. La demi-connexion envoyeur NE DEVRAIT PAS augmenter son taux d'envoi pendant environ un temps d'aller-retour après avoir vu un paquet avec une option Receveur lent. Après un temps d'aller-retour, l'effet du Receveur lent disparaît, permettant à la demi-connexion envoyeur d'augmenter son taux d'envoi. Par conséquent, la demi-connexion receveur DEVRAIT continuer à envoyer des options Receveur lent si elle a besoin à long terme d'empêcher la demi-connexion envoyeur d'aller plus vite. L'option Receveur lent n'indique pas d'encombrement, et la demi-connexion envoyeur n'a pas besoin de réduire son taux d'envoi. (Si nécessaire, le receveur peut forcer l'envoyeur à ralentir en éliminant des paquets, avec ou sans Données abandonnées, ou en rapportant de fausses déclarations de marque ECN.) Les API devraient laisser les applications receveuses établir l'option Receveur lent et les applications envoyeuses déterminent si leurs receveurs sont lents.

Receveur lent est une option d'un octet.



Receveur lent ne précise pas pourquoi le receveur a des difficultés à garder le rythme de l'envoyeur. Les raisons possibles incluent le manque d'espace de mémoire tampon, la surcharge du CPU, et des quotas d'application. Une application qui envoie pourrait réagir à Receveur lent en réduisant son taux d'envoi de niveau application, par exemple.

L'application envoyeuse ne doit cependant pas réagir à Receveur lent par l'envoi de davantage de données. Bien que la réponse optimale à un receveur limité en CPU puisse être de réduire la compression et d'envoyer plus de données (un format de données hautement compressées peut submerger un CPU lent plus sérieusement que ne le feraient les plus fortes exigences de mémoire d'un format de données moins compressées) ; ce genre de changement de format devrait être demandé au niveau des applications, et non pas via l'option Receveur lent.

Receveur lent met en œuvre une partie de la fonctionnalité de fenêtre de réception de TCP.

11.7 Option Données abandonnées

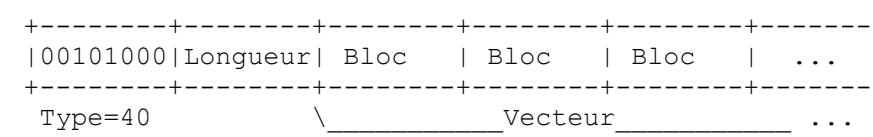
L'option Données abandonnées indique que les données d'application sur un ou plusieurs des paquets reçus n'ont pas en fait atteint l'application. Données abandonnées rapporte de plus pourquoi les données ont été supprimées : peut-être les données étaient corrompues, ou peut-être le receveur ne peut pas suivre le débit actuel de l'envoyeur et des données ont été

abandonnée dans certaines mémoires tampon de réception. En utilisation Données abandonnées, les points d'extrémité DCCP peuvent distinguer différents types de pertes ; ceci diffère de TCP, dans lequel toutes les pertes sont rapportées de la même manière.

Sauf spécification explicite contraire, les mécanismes de contrôle d'encombrement de DCCP DOIVENT réagir comme si chaque paquet Données abandonnées avait été marqué comme Encombrement ECN rencontré par le réseau. On a l'intention de permettre pour Données abandonnées des recherches plus riches sur les réponses d'encombrement aux paquets corrompus et autres abandonnés pour les points d'extrémité, mais les CCID DCCP DOIVENT réagir avec prudence à Données abandonnées jusqu'à ce comportement soit standardisé. Le paragraphe 11.7.2 décrit les réponses d'encombrement pour tous les codes d'abandon actuels.

Si les données d'applications d'un paquet reçu sont éliminées pour l'une des raisons énumérées ci-dessous, cela DEVRAIT être signalé en utilisant une option Données abandonnées. Autrement, le receveur PEUT choisir de ne déclarer comme "Reçu" que les paquets dont les données n'ont pas été abandonnées, sous réserve de la contrainte que les paquets non déclarés comme reçus NE DOIVENT PAS avoir eu leurs options traitées.

Les données de l'option ressemblent à ceci :



Le vecteur est constitué d'une série d'octets, appelés blocs, dont le codage de chacun correspond à l'un des deux choix :

Le premier octet de la première option Données abandonnées fait référence au paquet indiqué par le numéro d'accusé de réception ; les octets ultérieurs font référence à des paquets plus anciens. Données abandonnées NE DOIT PAS être envoyé sur des paquets DCCP-Données ou Demande-DCCP, qui n'ont pas de numéro d'accusé de réception, et toutes les options Données abandonnées reçues sur de tels paquets DOIVENT être ignorées.

Les blocs normaux, qui ont le bit de poids fort à 0, indiquent que tous les paquets reçus dans la plage en cours ont eu leurs données livrées à l'application. Les blocs abandonnés, qui ont le bit de poids fort à 1, indiquent que les paquets reçus dans la plage en cours n'ont pas été livrés comme d'habitude. Le champ Code d'abandon de 3 bits (DrpCd) dit ce qui s'est passé ; en général, aucune des données de ce paquet n'a atteint l'application. Les paquets signalés comme "Pas encore reçu" DOIVENT être inclus dans les blocs normaux ; les paquets non couverts par une option Données abandonnées sont traités comme si ils étaient dans un bloc normal. Les codes définis pour Abandon de blocs sont comme suit.

Code d'abandon	Signification
0	Contraintes du protocole
1	L'application n'écoute pas
2	Mémoire tampon de réception
3	Corrompu
4-6	Réservé
7	Livré corrompu

Tableau 7 : Codes d'abandon DCCP

Plus en détails :

0 Le paquet de données a été abandonné en raison des contraintes du protocole. Par exemple, les données ont été incluses dans un paquet Demande-DCCP, mais l'application de réception ne permet pas un tel portage, ou les données ont été incluses dans un paquet avec une Couverture de somme de contrôle anormalement basse.

1 Les données du paquet ont été abandonnées parce que l'application n'est plus à l'écoute. Voir le paragraphe 11.7.2.

2 Les données du paquet ont été abandonnées dans une mémoire tampon de réception, sans doute en raison d'un débordement de la mémoire tampon de réception. Voir le paragraphe 11.7.2.

3 Les données du paquet ont été abandonnées en raison de leur corruption. Voir le paragraphe 9.3.

7 Les données du paquet ont été corrompues, mais ont été quand même été livrées à l'application. Voir le paragraphe 9.3.

Par exemple, supposons qu'un paquet arrive avec le numéro d'accusé de réception 100, un vecteur d'accusé de réception rapporte que tous les paquets ont été reçus, et une option Données abandonnées contenant les valeurs décimales 0,160,3,162. Alors :
Le paquet 100 a été reçu (Numéro d'accusé de réception 100, bloc normal, longueur de plage 0).
Le paquet 99 a été abandonné dans une mémoire tampon de réception (bloc abandonné, code d'abandon 2, longueur de plage 0).
Les paquets 98, 97, 96 et 95 ont été reçus (bloc normal, longueur de plage 3).
Les paquets 95, 94, et 93 ont été abandonnés dans la mémoire tampon de réception (bloc abandonné, code d'abandon 2, longueur de plage 2).

Les longueurs de plage de plus de 128 (pour les blocs normaux) ou de 16 (pour les blocs abandonnés) doivent être codées sur plusieurs blocs. Une seule option Donnée abandonnées peut accuser réception de jusqu'à 32 384 blocs normaux de paquets de données, bien que le receveur NE DEVRAIT PAS envoyer une option Données abandonnées lorsque tous les paquets concernés tiennent dans les blocs normaux. Si plus de paquets devaient être acquittés qu'il ne peut en tenir dans 253 octets de Données abandonnées, plusieurs options Données abandonnées peuvent être envoyées. La deuxième option va commencer là où la première se termine, et ainsi de suite.

Une ou plusieurs options Données abandonnées qui, ensemble, rapportent l'état de plus de paquets qu'il n'en a été envoyé, ou qui changent le statut du paquet, ou qui sont en désaccord avec le Vecteur Ack ou les options équivalentes (en signalant un paquet "non encore reçu" comme "éliminé dans la mémoire tampon de réception", par exemple) DEVRAIENT être considérées comme invalides. Le DCCP receveur DEVRAIT soit ignorer ces options, soit répondre par la réinitialisation de la connexion avec le code de réinitialisation 5, "Erreur d'option".

Une interface d'application DCCP devrait laisser les applications receveuses spécifier les codes de d'abandon correspondant à des paquets reçus. Par exemple, cela permettrait aux applications de calculer leur propre somme de contrôle mais de rapporter encore des paquets "abandonnés à cause de corruption" via l'option Données abandonnées. L'interface NE DEVRAIT PAS laisser les applications réduire la "gravité" du code d'abandon d'un paquet ; par exemple, l'application ne devraient pas être en mesure de relever le niveau d'un paquet livré corrompu (code d'abandon 7) à livré normalement (pas de code d'abandon).

Les informations de données abandonnées sont transmises de manière fiable. C'est-à-dire, les points d'extrémité DEVRAIENT continuer de transmettre les options Données abandonnées jusqu'à réception d'un accusé de réception indiquant que les options pertinentes ont été traitées. En termes de Vecteur Ack, chaque accusé de réception devrait contenir des options Données abandonnées qui couvrent l'ensemble de la fenêtre d'accusé de réception (paragraphe 11.4.2), bien que lorsque tous les paquets dans cette fenêtre sont placés dans un bloc normal, aucune option réelle ne soit nécessaire.

11.7.1 Données abandonnées et réponse d'encombrement normale

Au moment de décider d'une réponse à un accusé de réception particulier ou à un ensemble d'accusés de réception contenant des options Données abandonnées, un mécanisme de contrôle d'encombrement DOIT considérer les paquets abandonnés, les marques ECN Encombrement rencontré (y compris les paquets marqués qui sont inclus dans les données abandonnées) et les paquets en évidence dans Données abandonnées. Pour les mécanismes fondés sur la fenêtre, l'espace de réponse valide est défini comme suit.

Supposons une vieille fenêtre W. On calcule indépendamment une nouvelle fenêtre W_n1 qui suppose qu'aucun paquet n'a été marqué Données abandonnées (donc W_n1 ne contient que la réponse normale d'encombrement) et une nouvelle fenêtre W_n2 qui suppose qu'aucun paquet n'a été perdu ou marqué (donc W_n2 ne contient que la réponse Données abandonnées). On suppose que Données abandonnées a recommandé une réduction de la fenêtre d'encombrement, de sorte que $W\_n2 < W$.

Puis la nouvelle fenêtre W_n réelle NE DOIT PAS être supérieure au minimum de W_n1 et W_n2, et l'envoyeur PEUT combiner les deux réponses, en réglant $W\_n = W + \min(W\_n1 - W, 0) + \min(W\_n2 - W, 0)$.

Les détails de la façon dont cela est accompli sont spécifiées dans les documents de profil de CCID. Les mécanismes de contrôle d'encombrement qui ne sont pas fondés sur la fenêtre DOIVENT se comporter de façon analogue ; là encore, les profils de CCID définissent comment.

11.7.2 Codes d'abandon particuliers

Le code d'abandon 0, Contraintes du protocole, n'indique aucune sorte d'encombrement, de sorte que le CCID envoyeur DEVRAIT réagir aux paquets qui ont le code d'abandon 0, comme si ils avaient été reçus (avec ou sans marques Encombrement ECN rencontré, comme approprié). Cependant, le point d'extrémité d'envoi NE DEVRAIT PAS envoyer de données jusqu'à ce qu'il estime que la contrainte du protocole ne s'applique plus.

Le code d'abandon 1, "L'application n'écoute pas", signifie que l'application qui s'exécute au point d'extrémité qui a envoyé l'option n'est plus à l'écoute des données. Par exemple, un serveur peut fermer sa demi-connexion receveur aux nouvelles données après avoir reçu une demande complète du client. Cela va limiter la quantité d'état disponible au niveau du serveur pour les données entrantes et donc réduire les dommages potentiels de certaines attaques de déni de service. Une option Données abandonnées contenant le code d'abandon 1 DEVRAIT être envoyée chaque fois que les données reçues sont ignorées en raison d'une application qui n'est pas à l'écoute. Quand un point d'extrémité rapporte un code d'abandon 1 pour un paquet, il DEVRAIT signaler le code d'abandon 1 pour chaque paquet de données suivant sur cette demi-connexion ; quand un point d'extrémité reçoit un rapport d'état d'abandon 1, il DEVRAIT s'attendre à ce qu'aucune autre donnée ne soit plus livrée à l'application de l'autre point d'extrémité, donc il NE DEVRAIT PAS envoyer d'autres données.

Le code d'abandon 2, Mémoire tampon de réception, indique l'encombrement à l'intérieur de l'hôte receveur. Par exemple, si une mémoire tampon de prise à abandon du noyau par la queue est trop pleine pour accepter les données d'application d'un paquet, ce paquet devrait être rapporté comme code d'abandon 2. Pour une mémoire tampon de prise à abandon par la tête ou plus complexe, le paquet abandonné devrait être signalé avec le code d'abandon 2. Les mises en œuvre DCCP peuvent aussi fournir une API permettant aux applications de marquer les paquets reçus avec le code d'abandon 2, indiquant que l'application n'a plus d'espace dans sa mémoire tampon de réception au niveau utilisateur. (Cependant, il n'est généralement pas utile de signaler que les paquets sont abandonnés suite à un code d'abandon 2 après que sont passés plus d'un couple de temps d'aller-retour. La demi-connexion envoyeur peut avoir à ce moment oublié son état d'accusé de réception pour le paquet, et donc le rapport de données abandonnées n'aura aucun effet.) Chaque paquet nouvellement acquitté comme code d'abandon 2 DEVRAIT réduire le taux instantané de l'envoyeur d'un paquet par temps d'aller-retour, à moins que l'envoyeur en soit déjà à envoyer un paquet par RTT, ou moins. Chaque profil de CCID définit le mécanisme spécifique de CCID par lequel cela est accompli.

Actuellement, les autres codes d'abandon (à savoir le code d'abandon 3, Corrompus, le code d'abandon 7, Livré corrompu, et les codes réservés 4 à 6) DOIVENT faire que le CCID pertinent se comporte comme si les paquets concernés étaient marqués ECN (Encombrement ECN rencontré).

12. Notification explicite d'encombrement

Le protocole DCCP est entièrement à capacité ECN *(Explicit Congestion Notification)* [RFC3168]. Chaque CCID spécifie comment ses points d'extrémité répondent aux marques ECN. Par ailleurs, DCCP, contrairement à TCP, permet aux envoyeurs de contrôler le débit auquel les accusés de réception sont générés (avec des options comme Taux d'accusés de réception) ; puisque l'encombrement des accusés de réception est contrôlé, ils sont également considérés comme un transport à capacité ECN.

Chaque profil de CCID décrit comment ce CCID interagit avec ECN, à la fois pour le trafic de données et pour le pur trafic d'accusés de réception. Un envoyeur DEVRAIT mettre Transport à capacité ECN sur ses en-têtes IP de paquets à moins que la caractéristique ECN-incapable du receveur soit établie ou que le CCID pertinent l'interdise.

Le reste de cette section décrit la caractéristique ECN-incapable et l'interaction du nom occasionnel *(nonce)* ECN avec des options d'accusé de réception telles que Vecteur d'accusé de réception.

12.1 Caractéristique ECN-Incapable

Les points d'extrémité DCCP sont à capacité ECN par défaut, mais la caractéristique ECN-Incapable permet à une extrémité de rejeter l'utilisation de la notification explicite d'encombrement. L'utilisation de cette caractéristique N'EST PAS RECOMMANDÉE. L'incapacité ECN évite à la fois les avantages possibles d'ECN et empêche les envoyeurs d'utiliser le nom occasionnel ECN pour vérifier l'inconduite du receveur. Une pile DCCP PEUT donc laisser inutilisée la caractéristique ECN-Incapable, agissant comme si toutes les connexions étaient à capacité ECN. Noter que les interactions inappropriées de pare-feu qui s'attachaient à la mise en œuvre ECN de TCP [RFC3360] impliquent les bits d'en-tête TCP, et non pas les bits ECN de l'en-tête IP ; on ne ECK aucun boîtier de médiation qui bloquerait les paquets DCCP à capacité ECN mais permettrait les paquets DCCP incapables d'ECN.

ECN-Incapable a le numéro de caractéristique 4 et est une priorité de serveur. Il prend des valeurs booléennes d'un octet. Le

DCCP A DOIT être capable de lire les bits ECN des en-têtes IP des trames reçues lorsque ECN-Incapable/A est zéro. (Ceci est indépendant de si il peut établir les bits ECN sur les trames envoyées.) Le DCCP A envoie donc une option "Change L(ECN-Incapable, 1)" à DCCP B pour l'informer que A ne peut pas lire les bits ECN. Si la caractéristique ECN-Incapable/A est un, alors tous les paquets de DCCP B DOIVENT être envoyés comme ECN-Incapable. Les nouvelles connexions commencent avec ECN-Incapable 0 (c'est-à-dire, à capacité ECN) pour les deux points d'extrémité. Les valeurs de deux ou plus sont réservées.

Si un DCCP n'est pas à capacité ECN, il DOIT envoyer les options obligatoires "Change L(ECN-Incapable, 1)" à l'autre extrémité jusqu'à leur acquittement (par "Confirme R(ECN-Incapable, 1)") ou que la connexion se ferme. Par ailleurs, il NE DOIT PAS accepter de données jusqu'à ce que l'autre extrémité envoie "Confirme R(ECN-Incapable, 1)". Il DEVRAIT envoyer des options Données abandonnées sur ses accusés de réception, avec le code d'abandon 0 ("Contraintes de protocole") si l'autre point d'extrémité envoie des données de façon inappropriée.

12.2 Noms occasionnels ECN

L'évitement d'encombrement ne se produira pas, et le receveur va parfois obtenir ses données plus rapidement, si l'envoyeur n'est pas informé des événements d'encombrement. Ainsi, le receveur a une certaine incitation à falsifier les information d'acquittement, en rapportant que des paquets marqués ou abandonnés ont été en fait reçus non marqués. Ce problème est plus grave avec DCCP qu'avec TCP, puisque TCP fournit un transport fiable : il est plus difficile avec le protocole TCP de mentir à propos des paquets perdus sans casser l'application.

Les noms occasionnels ECN sont un mécanisme général pour empêcher la tricherie sur ECN (ou une tricherie sur les pertes). Deux valeurs pour le champ de deux bits d'en-tête ECN indiquent le transport à capacité ECN, 01 et 10. Le second codet, 10, est le nom occasionnel ECN *(Nonce)*. En général, un envoyeur du protocole choisit au hasard entre ces codets sur ses paquets de sortie, se souvenant du numéro de séquence qu'il a choisi. Sur chaque accusé de réception, le receveur du protocole rapporte le numéro des noms occasionnels ECN qu'il a reçu jusqu'alors. C'est ce qu'on appelle l'écho de nom occasionnel ECN *(Nonce Echo)*. Comme le marquage ECN et l'abandon de paquet détruisent tous deux le nom occasionnel ECN, un receveur qui ment sur une marque ECN ou un abandon de paquet a une probabilité de 50 % de bien deviner en évitant la discipline. L'envoyeur peut réagir par des mesures punitives à une discordance de nom occasionnel ECN, peut-être jusqu'à abandonner la connexion. Le champ Écho de nom occasionnel ECN n'a pas besoin d'être un entier ; un bit est suffisant pour attraper 50 % des infractions, et la probabilité de réussite diminue de façon exponentielle à mesure que plus de paquets sont envoyés [RFC3540].

Dans DCCP, le champ Écho de nom occasionnel ECN est codé dans les options d'accusé de réception. Par exemple, l'option Vecteur Ack vient sous deux formes, Vecteur Ack [Nonce 0] (option 38) et Vecteur Ack [Nonce 1] (option 39), correspondant aux deux valeurs pour un écho de nom occasionnel ECN d'un bit. L'écho de nom occasionnel pour un Vecteur Ack donné est égal à la somme d'un bit (OU exclusif, ou parité) des noms occasionnels ECN pour les paquets rapportés par ce Vecteur Ack tels que reçus et non marqués ECN. Ainsi, seuls les paquets marqués comme État 0 importent pour ce calcul (c'est-à-dire, les paquets valides reçus qui n'ont pas été marqués ECN). Chaque option Vecteur Ack est suffisamment détaillée pour que l'envoyeur détermine ce que l'écho de nom occasionnel aurait dû être. Il peut vérifier ce calcul par rapport à l'écho de nom occasionnel réel et se plaindre si il y a une discordance. (Le Vecteur Ack pourrait éventuellement rendre compte de chaque état de nom occasionnel du paquet, mais cela limiterait sévèrement sa compressibilité, sans fournir beaucoup de protection supplémentaire.)

Chaque envoyeur DCCP DEVRAIT établir les noms occasionnels ECN sur ses paquets et se souvenir de quels paquets ont des noms occasionnels. Quand un émetteur détecte une discordance d'écho de nom occasionnel ECN, il se comporte comme décrit au paragraphe suivant. Chaque receveur DCCP DOIT calculer et utiliser la valeur correcte pour l'écho de nom occasionnel ECN lors de l'envoi des options d'accusé de réception.

L'incapacité ECN, comme indiqué par la caractéristique ECN-Incapable, est traitée comme suit : un point d'extrémité qui envoie des paquets à un receveur ECN incapable DOIT envoyer ses paquets comme ECN-Incapable, et un receveur ECN incapable DOIT utiliser la valeur zéro pour tous les échos de nom occasionnel ECN.

12.3 Répression des agressions

Les points d'extrémité DCCP ont plusieurs mécanismes pour la détection de mauvais comportements liés à l'encombrement. Par exemple :

- Un envoyeur peut détecter une discordance d'écho de nom occasionnel ECN, indiquant un possible mauvais comportement du receveur.
- Un receveur peut détecter si l'envoyeur répond aux rétroactions d'encombrement ou de receveur lent.
- Un point d'extrémité peut être capable de détecter que son homologue rapporte des petites valeurs de temps écoulé inappropriées (paragraphe 13.2).

Un point d'extrémité qui détecte un mauvais comportements possible lié à l'encombrement DEVRAIT essayer de vérifier

que son homologue se conduit vraiment mal. Par exemple, un point d'extrémité envoyeur pourrait envoyer un paquet dont le champ d'en-tête ECN est réglé à Encombrement rencontré, 11 ; un receveur qui ne rapporte pas une marque correspondante se comporte très probablement mal.

Lors de la détection d'une possible mauvaise conduite, un envoyeur DEVRAIT répondre comme si le receveur avait signalé un ou plusieurs paquets récents comme marqués ECN (au lieu de non marqués) tandis qu'un receveur DEVRAIT déclarer un ou plusieurs paquets récents non marqués comme des paquets marqués ECN. Autrement, un envoyeur peut agir comme si le receveur avait envoyé une option Receveur lent, et un receveur peut envoyer des options Receveur lent. D'autres réactions qui servent à ralentir le taux de transfert sont également acceptables. Une entité qui détecte de mauvais comportements particulièrement graves et continus PEUT également réinitialiser la connexion avec le code de réinitialisation 11, "Pénalité d'agression".

Cependant, les discordances de nom occasionnel ECN et autres signaux d'avertissement peuvent résulter de causes innocentes, telles que des erreurs de mise en œuvre ou des attaques. En particulier, une attaque réussie de DCCP-Données (paragraphe 7.5.5) peut amener le receveur à signaler un mauvais écho de nom occasionnel ECN. Par conséquent, la réinitialisation de la connexion et d'autres mécanismes lourds ne devraient être utilisés qu'en dernier ressort, après plusieurs temps d'aller-retour d'agression vérifiée.

13. Options de synchronisation

Les options Horodatage, Écho d'horodatage, et Temps écoulé aident les points d'extrémité DCCP à mesurer explicitement les temps d'aller-retour.

13.1 Option Horodatage

Cette option est permise sur n'importe quel paquet DCCP. La longueur de l'option est de 6 octets.

```
+--------+--------+--------+--------+--------+--------+
|00101001|00000110|      Valeur d'horodatage          |
+--------+--------+--------+--------+--------+--------+
Type = 41 Longueur = 6
```

Les quatre octets des données d'option portent l'horodatage de ce paquet. L'horodatage est un entier de 32 bits qui augmente de façon monotone avec le temps, à un taux de 1 unité par 10 microsecondes. À ce rythme, la valeur d'horodatage reviendra à zéro toutes les 11,9 heures. Les points d'extrémité n'ont pas besoin de mesurer le temps à cette fine granularité ; par exemple, un point d'extrémité qui préfère mesurer le temps à une granularité de millième de seconde pourrait envoyer des valeurs d'horodatage qui soient toutes des multiples de 100. Le moment précis correspondant à la valeur d'horodatage zéro n'est pas spécifié : les valeurs d'horodatage ne sont significative que par rapport à d'autres valeurs d'horodatage envoyées sur la même connexion. Un DCCP qui reçoit une option Horodatage DEVRAIT répondre avec une option Écho d'horodatage sur le prochain paquet qu'il envoie.

13.2 Option Temps écoulé

Cette option est permise sur n'importe quel paquet de DCCP qui contient un numéro d'accusé de réception ; de telles options reçues sur d'autres types de paquets DOIVENT être ignorées. Elle indique combien de temps s'est écoulé depuis que le paquet acquitté -- le paquet avec ce numéro d'accusé de réception -- a été reçu. L'option peut prendre 4 ou 6 octets, selon la taille de la valeur de Temps écoulé. Temps écoulé aide à corriger les estimations de temps d'aller-retour lorsque l'écart entre la réception d'un paquet et son accusé de réception peut être long -- dans CCID 3, par exemple, où les accusés de réception sont envoyés de façon peu fréquente.

```
+--------+--------+--------+--------+
|00101011|00000100|   Temps écoulé  |
+--------+--------+--------+--------+
Type = 43 Longueur = 4

+--------+--------+--------+--------+--------+--------+
|00101011|00000110|      Temps écoulé                 |
+--------+--------+--------+--------+--------+--------+
Type = 43 Longueur = 6
```

Les données d'option, Temps écoulé, représentent une estimation de la borne inférieure du temps écoulé depuis que le paquet acquitté a été reçu, avec des unités de centième de milliseconde. Si Temps écoulé est inférieur à une demi seconde, la première, plus petite forme de l'option DEVRAIT être utilisée. Temps écoulé de plus de 0,65535 secondes DOIT être envoyé en utilisant la deuxième forme de l'option. La valeur spéciale de Temps écoulé de 4 294 967 295, ce qui correspond à environ 11,9 heures, est utilisé pour représenter tout temps écoulé supérieur à 42 949,67294 secondes. Les points d'extrémité DCCP NE DOIVENT PAS rapporter des Temps écoulés qui soient significativement plus grands que le temps réellement passé. Une connexion PEUT être réinitialisé avec le code de réinitialisation 11, "Pénalité d'agression", si un point d'extrémité détermine que l'autre rapporte un temps écoulé beaucoup trop grand.

Le temps écoulé est mesuré en centièmes de milliseconde en tant que compromis entre deux objectifs contradictoires. Premièrement, cela fournit assez de granularité pour réduire les erreurs d'arrondi lors de la mesure du temps écoulé sur des réseaux locaux rapides ; deuxièmement, cela permet à de nombreux temps écoulés raisonnables de tenir sur deux octets de données.

13.3 Option Écho d'horodatage

Cette option est permise sur tout paquet DCCP, pour autant qu'au moins un paquet portant l'option Horodatage ait été reçu. Généralement, un point d'extrémité DCCP devrait envoyer une option Écho d'horodatage pour chaque option Horodatage qu'il reçoit, et il devrait envoyer cette option aussitôt qu'il peut. La durée de l'option est entre 6 et 10 octets, selon que le temps écoulé est inclus, et de combien il est.

```
+--------+--------+--------+--------+--------+--------+
|00101010|00000110|     Écho d'horodatage             |
+--------+--------+--------+--------+--------+--------+
Type = 42 Longueur = 6

+--------+--------+------- ... -------+--------+--------+
|00101010|00001000| Écho d'horodatage |  Temps écoulé   |
+--------+--------+------- ... -------+--------+--------+
Type = 42 Longueur= 8       (4 octets)

+--------+--------+------- ... -------+------- ... -------+
|00101010|00001010| Écho d'horodatage |    Temps écoulé   |
+--------+--------+------- ... -------+------- ... -------+
Type = 42 Longueur=10     (4 octets)           (4 octets)
```

Les quatre premiers octets des données d'option Écho d'horodatage portent une valeur d'horodatage prise d'une option Horodatage reçue précédemment. Habituellement, ce sera le dernier paquet reçu -- le paquet indiqué par le numéro d'accusé de réception, s'il en est un -- mais ce peut être un paquet précédant. Chaque horodatage reçu se traduit généralement par exactement un Écho d'horodatage transmis. Si un point d'extrémité a reçu plusieurs options Horodatage depuis la dernière fois qu'il a envoyé un paquet, il PEUT alors ignorer toutes les options Horodatage sauf celle incluse dans le paquet avec le plus grand numéro de séquence. Autrement, il PEUT inclure plusieurs options Écho d'horodatage dans sa réponse, chacune correspondant à une option d'horodatage différente.

La valeur de Temps écoulé, semblable à celle de l'option Temps écoulé, indique la quantité de temps écoulé depuis la réception du paquet dont on fait l'écho de l'horodatage. Ce temps DOIT avoir des unités de centième de milliseconde. Temps écoulé est destiné à aider l'envoyeur d'horodatage à séparer le temps aller-retour du réseau du temps de traitement du receveur de l'horodatage. Cela peut être particulièrement important pour les CCID dont les accusés de réception sont envoyés rarement, de sorte qu'il pourrait y avoir un retard considérable entre la réception d'une option Horodatage et l'envoi de l'écho d'horodatage correspondant. Un champ Temps écoulé manquant est équivalent à un Temps écoulé de zéro. La plus petite version d'option que peut contenir la valeur de temps écoulé pertinente DEVRAIT être utilisée.

14. Taille maximale de paquet

Une mise en œuvre DCCP DOIT maintenir la taille maximale des paquets (MPS, *maximum packet size*) autorisée pour chaque session DCCP active. La MPS est influencée par la taille maximum de paquet autorisée par le mécanisme actuel de contrôle de l'encombrement (CCMPS, *congestion control maximum packet size*), par la taille maximale de paquet prise en charge par les liaisons du chemin (PMTU, unité maximale de transmission du chemin) [RFC1191], et la longueur des en-têtes IP et DCCP.

Une interface d'application DCCP DEVRAIT laisser l'application découvrir la MPS actuelle de DCCP. Généralement, la mise en œuvre DCCP refusera d'envoyer des paquets plus gros que la MPS, retournant une erreur appropriée à l'application. Une interface DCCP PEUT permettre aux applications de demander la fragmentation des paquets plus grands que la PMTU, mais pas plus que la CCMPS. (Les paquets plus grands que la CCMPS DOIVENT être rejetés dans tous les cas.) La fragmentation NE DEVRAIT PAS être la valeur par défaut, car elle diminue la robustesse : un paquet entier est éliminé même si un seul de ses fragments est perdu. Les applications peuvent généralement obtenir une meilleure tolérance à l'erreur en produisant des paquets plus petits que la PMTU.

La MPS signalée à l'application DEVRAIT être influencée par la taille dont on s'attend à ce qu'elle soit exigée pour les en-têtes et options DCCP. Si l'application fournit des données qui, lorsqu'elles sont combinées avec les options que la mise en œuvre DCCP souhaite inclure, dépasseraient la MPS, la mise en œuvre devrait soit envoyer les options sur un paquet séparé (comme un DCCP-Ack) soit abaisser la MPS, éliminer des données, et retourner une erreur appropriée à l'application.

14.1 Mesure de la PMTU

Chaque point d'extrémité DCCP DOIT garder la trace de la PMTU en cours pour chaque connexion, sauf que ce n'est pas obligatoire pour les connexions IPv4 dont les applications ont demandé la fragmentation. La PMTU DEVRAIT être initialisée à partir de la MTU de l'interface qui sera utilisée pour envoyer les paquets. La MPS sera initialisée avec le minimum de la PMTU et de la CCMPS, si il en est.

La découverte classique de la PMTU utilise des paquets non fragmentables. Dans IPv4, ces paquets ont le bit IP "Ne pas fragmenter" (DF) établi ; dans IPv6, tous les paquets sont non fragmentables une fois émis par un hôte d'extrémité. Comme spécifié dans la [RFC1191], quand un routeur reçoit un paquet avec DF établi qui est plus grand que la MTU de la liaison suivante, il envoie en retour à la source un message ICMP "Destination injoignable" dont le code indique qu'un paquet non fragmentable était trop grand pour transmission (un message "Datagramme trop gros"). Quand une mise en œuvre DCCP reçoit un message Datagramme trop gros, elle diminue sa PMTU jusqu'à la valeur de la MTU du prochain bond donnée dans le message ICMP. Si la MTU donnée dans le message est zéro, l'envoyeur choisit une valeur pour la PMTU en utilisant l'algorithme décrit à la Section 7 de la [RFC1191]. Si la MTU donnée dans le message est supérieure à la PMTU actuelle, le message Datagramme trop gros est ignoré, comme décrit dans la [RFC1191]. (On est conscient que cela peut causer des problèmes pour les points d'extrémité DCCP derrière certains pare-feu.)

Une mise en œuvre DCCP peut permettre à l'application de demander à l'occasion que la découverte de la PMTU soit effectuée à nouveau. Cela réinitialisera la PMTU à la MTU de l'interface sortante. De telles demandes devraient être limitées en débit, par exemple, à une toutes les deux secondes.

Un émetteur DCCP PEUT traiter la réception d'un message ICMP Datagramme trop gros comme une indication que le paquet signalé n'a pas été perdu en raison de l'encombrement, et donc, pour les besoins du contrôle d'encombrement, il PEUT ignorer l'indication du receveur DCCP que ce paquet n'est pas arrivé. Toutefois, si il fait cela, l'envoyeur DCCP DOIT alors vérifier les bits ECN de l'en-tête IP en écho dans le message ICMP et n'effectuer cette optimisation que si ces bits ECN indiquent que le paquet n'a pas rencontré d'encombrement avant d'atteindre le routeur dont la MTU de chemin est dépassée.

Une mise en œuvre DCCP DEVRAIT s'assurer, autant que possible, que les messages ICMP "Datagramme trop gros" étaient réellement générés par les routeurs, afin que des attaquants ne puissent pas réduire la PMTU à une valeur faussement petite. La manière la plus simple de le faire est de vérifier que le numéro de séquence sur l'en-tête encapsulé d'erreur ICMP correspond à un numéro de séquence que la mise en œuvre a récemment envoyé. (Selon les spécifications actuelles, les routeurs devraient retourner l'en-tête DCCP complet et la charge utile jusqu'à un maximum de 576 octets [RFC1812] ou la MTU IPv6 minimum [RFC2463], bien qu'ils ne soient pas tenus de retourner plus de 64 bits [RFC0792]. Tout montant supérieur à 128 bits comprendra le numéro de séquence.) Les messages ICMP Datagramme trop gros avec des numéros de séquence erronés ou manquants peuvent être ignorés, ou la mise en œuvre DCCP peut diminuer seulement temporairement la PMTU en réponse. Cependant, si plus de trois messages Datagramme trop gros étranges sont reçus et si l'autre point d'extrémité DCCP rapporte plus de trois paquets perdus, la mise en œuvre DCCP DEVRAIT supposer la présence d'un routeur confus et, soit respecter la PMTU des messages ICMP, soit (sur les réseaux IPv4) passer à permettre la fragmentation.

DCCP permet également le sondage de la PMTU vers l'avant [RFC4821], où le point d'extrémité DCCP commence par l'envoi de petits paquets avec DF établi puis augmente progressivement la taille des paquets jusqu'à ce qu'un paquet soit perdu. Ce mécanisme n'exige aucun traitement d'erreur ICMP. Les paquets DCCP-Sync sont le meilleur choix pour sonder vers l'avant, puisque les sondes DCCP-Sync ne risquent pas de perte de données de l'application. La mise en œuvre DCCP insère des données arbitraires dans la zone d'application de DCCP-Sync, en bourrant le paquet à la bonne longueur. Puisque chaque DCCP-Sync valable génère un DCCP-SyncAck immédiat en réponse, le point d'extrémité va avoir une assez bonne idée du moment où une sonde est perdue.

14.2 Comportement de l’envoyeur

Un émetteur DCCP DEVRAIT envoyer tous les paquets comme non fragmentables, comme décrit ci-dessus, avec les exceptions suivantes.

- Sur les connexions IPv4 dont l'application a demandé la fragmentation, l'envoyeur DEVRAIT envoyer les paquets avec le bit DF non établi.

- Sur les connexions IPv6 dont les applications ont demandé la fragmentation, l'envoyeur DEVRAIT utiliser les en-têtes d'extension de fragmentation pour fragmenter les paquets plus grands que la PMTU en tronçons de taille convenable. (Ces tronçons sont, bien sûr, non fragmentables.)

- Il n'est pas souhaitable que la découverte de la PMTU se produise sur la prise de contact initiale d'établissement de la connexion, car le processus d'établissement de la connexion peut ne pas être représentatif de la taille des paquets utilisés lors de la connexion, et l'exécution de la découverte de la MTU sur la prise de contact initiale peut retarder inutilement l'établissement de la connexion. Donc, les paquets Demande-DCCP et DCCP-Réponse DEVRAIENT être envoyés comme fragmentables. De plus, les paquets DCCP-Reset DEVRAIENT être envoyés comme fragmentables, bien que généralement ceux-ci soient assez petits pour ne pas être un problème. Pour les connexions IPv4, ces paquets DEVRAIENT être envoyés avec le bit DF bit non établi ; pour les connexions IPv6, ils DEVRAIENT être préventivement fragmentés à une taille ne dépassant pas la MTU de l’interface pertinente.

Si la mise en œuvre DCCP a diminué la PMTU, si l'application envoyeuse n'a pas demandé la fragmentation, et si l'application envoyeuse tente d'envoyer un paquet plus grand que la nouvelle MPS, l'API DOIT refuser d'envoyer le paquet et retourner une erreur approprié à l'application. L'application devrait alors utiliser l'API pour demander la nouvelle valeur de MPS. Il peut y avoir des paquets en mémoire tampon en attente de transmission qui sont plus petits que la vieille MPS mais plus grands que la nouvelle. Ces paquets PEUVENT être envoyés comme fragmentables, ou PEUVENT être éliminés ; ils NE DOIVENT PAS être envoyés comme non fragmentables.

15. Compatibilité vers l'avant

Des versions futures de DCCP pourront ajouter de nouvelles options et caractéristiques. Quelques lignes directrices simples permettront aux DCCP étendus d'interopérer avec les DCCP normaux.

- Les processeurs DCCP NE DOIVENT PAS agir de façon punitive à l'égard des options et caractéristiques qu'ils ne connaissent pas. Par exemple, les processeurs DCCP NE DOIVENT PAS réinitialiser la connexion si un certain champ marqué réservé dans la présente spécification est non nul, si une option inconnue est présente, ou si certaines options de négociation de caractéristique mentionnent une caractéristique inconnue. Au lieu de cela, les processeurs DCCP DOIVENT ignorer ces événements. L'option Obligatoire est la seule exception : si Obligatoire précède certaines options ou caractéristiques inconnues, la connexion doit être réinitialisée.

- Les processeurs DCCP DOIVENT anticiper la possibilité de valeurs inconnues d'une caractéristique, qui pourraient survenir dans le cadre d'une négociation pour une caractéristique connue. Pour les caractéristiques à priorité de serveur, les valeurs inconnues sont traitées comme normalement : comme la liste des priorités du DCCP non étendu ne contiendra pas de valeurs inconnues, le résultat de la négociation ne peut pas être une valeur inconnue. Un DCCP DOIT répondre par une option Confirme vide si elle contient une valeur inacceptable pour certaines caractéristiques non négociables.

- Chaque extension DCCP DEVRAIT être contrôlée par une caractéristique. La valeur par défaut de cette caractéristique DEVRAIT correspondre à "extension non disponible". Si un DCCP étendu veut utiliser l'extension, il DEVRAIT tenter de changer la valeur de la caractéristique en utilisant une option Change L ou Change R. Tout DCCP non étendu va ignorer l'option, laissant donc la caractéristique à sa valeur par défaut, "Extension non disponible".

La Section 19 fait la liste des numéros alloués à DCCP réservés à des fins expérimentales et d'essais.

16 Considérations sur les boîtiers de médiation

Cette section décrit les propriétés de DCCP que les pare-feu, les traducteurs d'adresses réseau, et autres boîtiers de médiation devraient considérer, y compris les parties du paquet que les boîtiers de médiation ne devraient pas changer.

L'intention est d'attirer l'attention sur les aspects de DCCP qui peuvent être utiles, ou dangereux, pour les boîtiers de médiation, ou qui diffèrent significativement de TCP.

Le champ Code de service dans les paquets Demande-DCCP fournissent des informations qui peuvent être utiles pour les boîtiers de médiation à états pleins. Avec le code de service, un boîtier de médiation peut dire quel protocole va utiliser une connexion sans s'appuyer sur les numéros d'accès. Les boîtiers de médiation peuvent interdire les connexions qui tentent d'accéder à des services inattendus en envoyant un DCCP-Reset avec le code de réinitialisation 8, "Mauvais code de service". Les boîtiers de médiation ne devraient pas modifier le code de service sauf si ils changent réellement le service auquel accède la connexion.

Les champs Accès de source et de destination sont dans les mêmes localisations de paquet que les champs correspondants dans TCP et UDP, ce qui peut simplifier certaines mises en œuvre de boîtiers de médiation.

Les considérations de compatibilité vers l'avant de la Section 15 s'appliquent aussi aux boîtiers de médiation. En particulier, les boîtiers de médiation ne devraient généralement pas agir de façon punitive envers les options et les caractéristiques qu'ils ne comprennent pas.

La modification des numéros de séquence et des numéros d'accusé de réception DCCP est plus ennuyeuse et dangereuse que la modification des numéros de séquence TCP. Un boîtier de médiation qui a ajouté ou retiré des paquets d'une connexion DCCP aurait à modifier au minimum les options d'accusé de réception, telles que Vecteur Ack, et les options spécifiques du CCID, telles que Intervalles de perte du TFRC. Sur les connexions à capacité ECN, le boîtier de médiation aurait à garder trace des informations de nom occasionnel ECN pour les paquets qu'il a introduits ou supprimés, de façon à ce que les options pertinentes d'accusé de réception continuent d'avoir des échos de nom occasionnel ECN corrects, sous peine de risquer que la connexion soit réinitialisée pour "Pénalité d'agression". On recommande donc que les boîtiers de médiation ne modifient pas les flux de paquets en ajoutant ou en supprimant des paquets.

Noter qu'il y a moins besoin de modifier les numéros de séquences DCCP par paquet que de modifier les numéros de séquence TCP par octet ; par exemple, un boîtier de médiation peut changer le contenu d'un paquet sans changer son numéro de séquence. (Dans TCP, la modification du numéro de séquence est nécessaire pour prendre en charge des protocoles, tels que FTP, qui portent des adresses à longueur variable dans le flux de données. Si une telle application était déployée sur DCCP, les boîtiers de médiation augmenteraient ou diminueraient simplement les paquets pertinents comme nécessaire sans changer leur numéro de séquence. Cela pourrait impliquer de fragmenter le paquet.)

Les boîtiers de médiation peuvent, bien sûr, réinitialiser les connexions en cours. De toute évidence, cela nécessite l'insertion d'un paquet dans un ou deux flux de paquets, mais ne pose pas de problèmes difficiles.

DCCP est peu favorable à "l'épissage de connexion" [SHHP00], dans lequel les tentatives de connexion des clients sont interceptées, et éventuellement ensuite "épissés" avec des connexions à un serveur externe via des manipulations de numéro de séquence. Un épisseur de connexion aurait au minimum à s'assurer que les connexions épissées sont d'accord sur toutes les valeurs de caractéristiques pertinentes, ce qui pourrait entraîner quelques renégociation.

Le contenu de cette section ne devrait pas être interprété comme une approbation sans réserve des boîtiers de médiation à états pleins.

17. Relations avec d'autres spécifications

17.1. RTP

Le protocole de transport en temps réel (RTP, *Real-Time Transport*) [RFC3550] est actuellement utilisé sur UDP par de nombreuses applications cibles de DCCP (par exemple, les supports de flux continu). Par conséquent, il est important d'examiner les relations entre DCCP et RTP et, en particulier, la question de savoir si des changements sont nécessaires ou souhaitables dans RTP quand il est mis en couche par dessus DCCP au lieu de UDP.

Il y a deux sources potentielles de redondance dans la combinaison RTP par dessus DCCP : les informations d'acquittement dupliquées et les numéros de séquence dupliqués. Ensemble, ces sources de redondance ajoutent un peu plus de 4 octets par paquet par rapport à RTP par dessus UDP, et éliminer la redondance ne réduirait pas le supplément de charge.

Tout d'abord, considérons les accusés de réception. RTP et DCCP rapportent tous deux la rétroaction sur les taux de perte pour les envoyeurs de données, via l'envoyeur de protocole de contrôle RTP et les rapports de receveur (paquets RTCP SR/RR) et via les options d'accusé de réception DCCP. Ces mécanismes de rétroaction sont potentiellement redondants. Toutefois, les paquets RTCP SR/RR contiennent des informations non présentes dans les accusés de réception DCCP, tels

que "gigue inter arrivée", et les accusés de réception de DCCP contiennent des informations non transmises par RTCP, telles que l'écho de nom occasionnel ECN. Aucun de ces mécanismes de rétroaction ne rend l'autre redondant.

L'envoi des deux types de rétroactions n'a cependant pas besoin d'être particulièrement coûteux. Les rapports RTCP peuvent être envoyés relativement rarement : une fois toutes les 5 secondes en moyenne, pour des flux à faible bande passante. Dans DCCP, certains mécanismes de rétroaction sont coûteux -- Vecteur Ack, par exemple, est fréquent et verbeux -- mais d'autres sont relativement bon marché : les accusés de réception CCID 3 (TFRC) prennent entre 16 et 32 octets d'options envoyés une fois par temps d'aller-retour. (Un rapport moins fréquent qu'une fois par RTT rendrait le contrôle d'encombrement moins sensible à la perte.) On conclura donc que les frais généraux d'accusé de réception dans RTP sur DCCP ne doivent pas être significativement plus élevés que pour RTP sur UDP, au moins pour CCID 3.

Une redondance claire peut être abordée au niveau de l'application. Les rapports verbeux de perte paquet par paquet envoyés dans les blocs RLE de rapports de perte étendus de RTCP [RFC3611] peuvent être déduits des options Vecteur Ack de DCCP. (L'inverse n'est pas vrai, car les blocs RLE de perte ne contiennent pas d'informations ECN.) Comme les mises en œuvre de DCCP devraient fournir une API pour l'accès des applications aux informations de Vecteur Ack, les applications RTP sur DCCP peuvent demander soit des Vecteurs Ack DCCP, soit des blocs RLE de rapport de perte de RTCP étendu, mais pas les deux.

Considérons maintenant la redondance de numéro de séquence sur les paquets de données. L'en-tête RTP intégré contient un numéro de séquence RTP de16 bits. La plupart des paquets de données vont utiliser le type DCCP-Data ; les paquets DCCP-DataAck et DCCP-Ack ne doivent généralement pas être envoyés. L'en-tête DCCP-Data est de 12 octets sans options, incluant un numéro de séquence de 24 bits. C'est 4 octets de plus qu'un en-tête UDP. Toutes les options nécessaires sur les paquets de données vont ajouter encore des frais généraux, bien que de nombreux CCID (par exemple, CCID 3, TFRC) ne nécessitent pas d'option sur la plupart des paquets de données.

Le numéro de séquence DCCP ne peut pas être déduit du numéro de séquence RTP car il s'incrémente sur les paquets non de données aussi bien que sur les paquets de données. Le numéro de séquence RTP ne peut pas être déduit non plus du numéro de séquence DCCP [RFC3550]. Par ailleurs, la suppression du numéro de séquence RTP n'économiserait pas d'espace d'en-tête en raison des questions d'alignement. On recommande donc que RTP transmis sur DCCP utilise les mêmes en-têtes que défini actuellement. Le coût de 4 octets de l'en-tête est un compromis raisonnable pour les caractéristiques de contrôle d'encombrement de DCCP et l'accès à ECN. Les points d'extrémité vraiment en manque de bande passante devraient utiliser des schémas de compression d'en-tête.

17.2 Gestionnaire d'encombrement et multiplexage

Comme DCCP ne fournit pas une livraison ordonnée fiable, plusieurs sous flux d'application peuvent être multiplexés sur une seule connexion DCCP sans pénalité de performance inhérente. Ainsi, il n'est pas besoin que DCCP fournisse un support intégré pour les multiples sous flux. Cela diffère de SCTP [RFC2960].

Certaines applications pourraient vouloir partager l'état de contrôle d'encombrement entre plusieurs flux DCCP qui partagent la même adresse de source et de destination. Cette fonctionnalité pourrait être fournie par le gestionnaire d'encombrement [RFC3124], une facilité générique de multiplexage. Toutefois, le gestionnaire d'encombrement ne prendrait pas totalement en charge DCCP sans changement ; par exemple, il ne gère pas facilement plusieurs mécanismes de contrôle d'encombrement.

18. Considérations sur la sécurité

DCCP n'offre pas de garantie de sécurité cryptographique. Les applications qui désirent des services de sécurité cryptographique (intégrité, authentification, confidentialité, contrôle d'accès et protection anti répétition) devraient utiliser IPsec ou une sorte de sécurité de bout en bout ; RTP sécurisé est un protocole candidat [RFC3711].

Néanmoins, DCCP est destiné à protéger contre certaines classes d'attaquants : les attaquants ne peuvent pas pirater une connexion DCCP (fermer la connexion de manière inattendue, ou faire que des données de l'attaquant soient acceptées par un point d'extrémité comme si elles provenaient de l'envoyeur) sauf si ils peuvent deviner des numéros de séquence valides. Donc, tant que les points d'extrémité choisissent bien leur numéro de séquence initial, un attaquant DCCP doit espionner les paquets de données pour obtenir une probabilité raisonnable de succès. La vérification de la validité des numéros de séquence fournit cette garantie. Le paragraphe 7.5.5 décrit en détails la sécurité du numéro de séquence. Cette propriété de sécurité ne tient qu'en supposant des nombres aléatoires de DCCP choisis selon les directives de la [RFC4086].

DCCP fournit également des mécanismes pour limiter l'impact potentiel de certaines attaques de déni de service. Ces

mécanismes comprennent Init Cookie (paragraphe 8.1.4), le paquet DCCP-CloseReq (paragraphe 5.5), le code d'abandon "L'application n'écoute pas" (paragraphe 11.7.2), les limitations au traitement des options qui pourraient provoquer la réinitialisation de connexion (paragraphe 7.5.5), les limitations sur le traitement de certains messages ICMP (paragraphe 14,1), et diverses limites de taux, ce qui permet aux serveurs d'éviter de gros calculs ou génération de paquets (paragraphes 7.5.3, 8.1.3, et autres).

DCCP ne fournit aucune protection contre les attaquants qui peuvent espionner les paquets de données.

18.1 Considérations de sécurité pour les sommes de contrôle partielles

La facilité de somme de contrôle partielle a un impact de sécurité distinct, notamment dans son interaction avec les mécanismes d'authentification et de chiffrement. L'impact est le même dans DCCP et le protocole UDP-Lite, et ce qui suit est une adaptation du texte correspondant dans la spécification UDP-Lite [RFC3828].

Lorsque le champ Couverture de somme de contrôle d'un paquet DCCP n'est pas zéro, la partie non couverte d'un paquet peut changer dans le transit. Ceci est contraire à l'idée derrière la plupart des mécanismes d'authentification : l'authentification réussit si le paquet n'a pas changé dans le transit. Sauf si les mécanismes d'authentification qui opèrent uniquement sur la partie sensible des paquets sont développés et utilisés, l'authentification va toujours échouer pour les paquets DCCP à somme de contrôle partielle dont la partie non couverte a été endommagée.

Le contrôle d'intégrité IPsec (ESP, Encapsulation Security Protocol, ou AH, Authentication Header) est appliqué (au moins) à la charge utile entière du paquet IP. La corruption de tout bit dans cette zone va alors entraîner l'élimination du paquet DCCP par le receveur IP, même si la corruption s'est produite dans une partie non couverte des données de l'application DCCP.

Lorsque IPsec est utilisé avec le chiffrement de la charge utile ESP, une liaison ne peut pas déterminer le protocole de transport spécifique d'un paquet transmis en inspectant la charge utile du paquet IP. Dans ce cas, la liaison DOIT fournir un contrôle d'intégrité standard couvrant l'ensemble du paquet IP et sa charge utile. Les sommes de contrôle DCCP partielles ne fournissent aucun avantage dans ce cas.

Le chiffrement (par exemple, au niveau du transport ou de l'application) peut être utilisé. Noter que l'omission de la vérification de l'intégrité peut, dans certaines circonstances, [B98] compromettre la confidentialité.

Si quelques bits d'un paquet chiffré sont endommagés, la transformation de déchiffrement va généralement étendre les erreurs de telle sorte que le paquet devient trop endommagé pour être utilisé. Beaucoup de transformations de chiffrement d'aujourd'hui présentent ce comportement. Il existe des transformations de chiffrement, les chiffrements par flux, qui ne causent pas la propagation des erreurs. L'utilisation appropriée des chiffrements de flux peut être assez difficile, surtout lorsque la vérification de l'authentification est omise [BB01]. En particulier, un attaquant peut provoquer des changements prévisibles au dernier texte en clair, même sans être capable de déchiffrer le texte chiffré.

19. Considérations relatives à l'IANA

L'IANA a attribué le numéro de protocole IP 33 à DCCP.

DCCP présente huit séries de nombres dont les valeurs doivent être allouées par l'IANA. On se réfère aux politiques d'allocation, comme l'action de normalisation, décrites dans la [RFC2434], et la plupart des registres réservent certaines valeurs à des fins expérimentales et d'essais [RFC3692]. En outre, DCCP exige que le registre des numéros d'accès de l'IANA soit ouvert pour les enregistrements d'accès DCCP ; le paragraphe 19.9 décrit comment. L'IANA devrait avoir toute liberté pour contacter l'expert réviseur de DCCP pour les questions sur tout registre, indépendamment de la politique du registre, pour toute clarification ou si il y a un problème avec une demande.

19.1 Registre des types de paquet

Chaque entrée dans le registre Types de paquet DCCP contient un type de paquet, qui est un nombre dans la gamme 0-15, un nom de type de paquet, comme Demande-DCCP, et une référence à la RFC définissant le type de paquet. Le registre est d'abord rempli en utilisant les valeurs du tableau 1 (paragraphe 5.1). Ce document alloue les types de paquets 0-9, et le type de paquet 14 est en permanence réservé pour une utilisation expérimentale et d'essais. Les types de paquets 10-13 et 15 sont actuellement réservés et devraient être alloués selon la politique d'action de normalisation, ce qui nécessite l'examen et l'approbation de l'IESG et la publication d'une RFC de l'IETF sur la voie de la normalisation.

19.2 Registre des codes de réinitialisation

Chaque entrée dans le registre des codes de réinitialisation DCCP contient un code de réinitialisation, qui est un nombre dans la gamme 0-255, une courte description du code de réinitialisation, telle que "Pas de connexion", et une référence à la RFC qui définit le code de réinitialisation. Le registre est rempli initialement à l'aide les valeurs du tableau 2 (paragraphe 5.6). Ce document alloue les codes de réinitialisation 0-11, et les codes de réinitialisation de 120-126 sont en permanence réservés aux utilisations expérimentales et d'essais. Les codes de réinitialisation de 12-119 et 127 sont actuellement réservés et devraient être alloués selon la politique de consensus de l'IETF, nécessitant une publication de RFC de l'IETF (sur la voie de la normalisation ou pas) avec examen et approbation de l'IESG. Les codes de réinitialisation 128-255 sont en permanence réservés aux registres spécifiques de CCID ; chaque document de profil de CCID décrit comment le registre correspondant est géré.

19.3 Registre des types d'option

Chaque entrée dans le registre des types d'option DCCP contient un type d'option, qui est un nombre dans la gamme 0-255, le nom de l'option, telle que "Receveur lent", et une référence à la RFC définissant le type d'option. Le registre est initialement rempli en utilisant les valeurs du tableau 3 (paragraphe 5.8). Ce document alloue les types d'options 0-2 et 32-44, et les types d'option de 31 et 120 à 126 sont réservés aux utilisations expérimentales et d'essais. Les types d'option 3-30, 45-119, et 127 sont actuellement réservés et devraient être alloués selon la politique de consensus de l'IETF, nécessitant une publication de RFC de l'IETF (sur la voie de la normalisation ou pas) avec examen et approbation de l'IESG. Les types d'option de 128 à 255 sont réservés aux registres spécifiques des CCID ; chaque document de profil de CCID décrit comment le registre correspondant est géré.

19.4 Registre des numéros de caractéristique

Chaque entrée dans le registre des numéros de caractéristique DCCP contient un numéro de caractéristique, qui est un nombre dans la gamme 0-255, le nom de la caractéristique, tel que "ECN incapable", et une référence à la RFC définissant le numéro de caractéristique. Le registre est initialement rempli en utilisant les valeurs du tableau 4 (Section 6). Ce document alloue les numéros de caractéristique 0 à 9, et les numéros de caractéristique de 120 à 126 sont réservés aux utilisations expérimentales et d'essais. Les numéros de caractéristique 10-119 et 127 sont actuellement réservés et devraient être alloués selon la politique de consensus de l'IETF, nécessitant une publication de RFC de l'IETF (sur la voie de la normalisation ou pas) avec examen et approbation de l'IESG . Les numéros de caractéristique128-255 sont réservés aux registres spécifiques des CCID, chaque document de profil de CCID décrit comment le registre correspondant est géré.

19.5 Registre des identifiants de contrôle d’encombrement

Chaque entrée dans le registre des identifiants de contrôle d’encombrement DCCP (CCID) contient un CCID, qui est un nombre dans la gamme 0-255, le nom du CCID, tel que "contrôle d'encombrement de type TCP", et une référence à la RFC définissant le CCID. Le registre est initialement rempli en utilisant les valeurs du tableau 5 (Section 10). Les CCID 2 et 3 sont alloués par les profils publiés simultanément, et les CCID 248-254 sont réservés à un usage expérimental et d'essai. Les CCID 0, 1, 4-247, et 255 sont actuellement réservés et devraient être alloués selon la politique de consensus de l'IETF, ce qui nécessite la publication d'une RFC de l'IETF (sur la voie de la normalisation ou non) avec examen et approbation de l'IESG.

19.6 Registre des états de vecteur d’accusé de réception

Chaque entrée dans le registre des états de Vecteur Ack DCCP contient un état de Vecteur Ack, qui est un nombre dans la gamme 0-3; le nom de l'état, tel que "Reçu marqué ECN", et une référence à la RFC qui définir l'état. Le registre est rempli initialement en utilisant les valeurs du tableau 6 (paragraphe 11.4). Ce document alloue les états 0, 1 et 3. L'état 2 est actuellement réservé et devrait être alloué selon la politique d'action de normalisation, qui nécessite l'examen et l'approbation de l'IESG et la publication d'une RFC de l'IETF sur la voie de la normalisation.

19.7 Registre des codes d’abandon

Chaque entrée dans le registre des codes d'abandon DCCP contient un code d'abandon de données éliminées, qui est un nombre dans la gamme 0-7, le nom du code d'abandon, tel que "L'application n'est pas à l'écoute", et une référence à la RFC qui définit le code d'abandon. Le registre est d'abord rempli à l'aide des valeurs du tableau 7 (paragraphe 11.7). Ce document alloue les codes d'abandon 0-3 et 7. Les codes d'abandon 4-6 sont actuellement réservés, et devraient être alloués selon la politique d'action de normalisation, qui exige l'examen et l'approbation de l'IESG et la publication d'une RFC de l'IETF sur la voie de la normalisation.

19.8 Registre des codes de service

Chaque entrée dans le registre des codes de service contient un code de service, qui est un nombre dans la gamme 0 à 4 294 967 294; une courte description du service prévu; une référence facultative à une RFC ou autre spécification publiquement disponible, définissant le code de service. Le registre devrait indiquer la valeur numérique du code de service comme un nombre décimal. Lorsque le code de service peut être représenté en format "SC:" selon les règles du paragraphe 8.1.2, le registre devrait également montrer l'interprétation ASCII correspondante du Code de service moins le préfixe "SC:". Ainsi, le nombre 1 717 858 426 devrait de plus apparaître comme "fdpz". Les codes de service ne sont pas spécifiques de DCCP. Le code de service 0 est réservé (il représente l'absence de code de service significatif), et les codes de service 1 056 964 608 à 1 073 741 823 (octet ASCII haut "?") sont réservés à un usage privé. Noter que 4 294 967 295 n'est pas un code de service valide. La plupart des autres codes de service sont alloués au premier arrivé, premier servi, sans publication de RFC requise ; les exceptions sont énumérées au paragraphe 8.1.2. Ce document alloue un code de service unique, 1 145 656 131 ("DISC"). Cela correspond au service d'abandon, qui annule toutes les données envoyées au service et n'envoie pas de données en réponse.

19.9 Registre des numéros d’accès

Les services DCCP peuvent utiliser des numéros d'accès de contact pour fournir des services à des appelants inconnus, comme dans TCP et UDP. Il est donc demandé à l'IANA d'ouvrir le registre existant des numéros d'accès pour DCCP en utilisant les règles suivantes, qu'on a l'intention de mailler avec les procédures d'enregistrement de numéros d'accès existantes.

Les numéros d'accès sont divisés en trois gammes. Les accès bien connus sont ceux de 0 à 1023, les accès enregistrés sont ceux de 1024 à 49 151, et les accès dynamiques et/ou privés sont ceux de 49 152 à 65 535. Les accès bien connus et enregistrés sont destinés à être utilisés par les applications serveur qui désirent un point de contact par défaut sur un système. Sur la plupart des systèmes, les accès bien connus ne peuvent être utilisés que par les processus système (ou racine) ou par des programmes exécutés par les utilisateurs privilégiés, alors que les accès enregistrés peuvent être utilisés par les processus utilisateur ordinaires ou des programmes exécutés par des utilisateurs ordinaires. Les accès dynamiques et/ou privés sont destinés à un usage temporaire, y compris les accès côté client, les accès négociés hors bande, et l'essai d'application avant l'enregistrement d'un accès dédié ; ils NE DOIVENT PAS être enregistrés.

Le registre des numéros d'accès devrait accepter les enregistrements pour les accès DCCP dans les gammes des accès bien connus et des accès enregistrés. Les accès bien connus et les accès enregistrés NE DEVRAIT PAS être utilisés sans enregistrement. Bien que dans certains cas -- comme le portage d'une application de UDP à DCCP -- il puisse sembler naturel d'utiliser un accès DCCP avant l'achèvement de l'enregistrement, on souligne que l'IANA ne garantit pas l'enregistrement des accès bien connus et enregistrés particuliers . Les enregistrements devraient être demandés aussitôt que possible.

Chaque enregistrement d'accès DOIT inclure les informations suivantes :
- o Un court nom de l'accès, entièrement composé de lettres (A à Z et a à z), de chiffres (0 à 9), et des caractères de ponctuation à partir de "-_+./*" (non compris les guillemets).
- o Le numéro d'accès dont l'enregistrement est demandé.
- o Une courte phrase en anglais décrivant l'objet de l'accès. Elle DOIT inclure un ou plusieurs descripteurs de code de service textuels séparés par des espaces désignant les codes de service correspondants de l'accès (voir au paragraphe 8.1.2).
- o Les informations de nom et contact de la personne ou entité effectuant l'enregistrement et, éventuellement, une référence à un document définissant l'utilisation de l'accès. Les enregistrements provenant de groupes de travail de l'IETF doivent seulement nommer le groupe de travail, mais indiquer une personne de contact est recommandé.

Les déclarants sont invités à suivre ces directives lors de la soumission d'un enregistrement :
- o Un nom d'accès NE DEVRAIT PAS être enregistré pour plus d'un numéro d'accès DCCP.
- o Un nom d'accès enregistré pour UDP PEUT aussi être enregistré pour DCCP. Un tel enregistrement DEVRAIT utiliser le même numéro d'accès que l'enregistrement UDP existant.
- o L'intention concrète d'utiliser un accès DEVRAIT précéder l'enregistrement de l'accès. Par exemple, les accès UDP existants NE DEVRAIENT PAS être enregistrés avant toute intention d'utiliser ces accès pour DCCP.
- o Un nom d'accès généralement associé à TCP et/ou SCTP NE DEVAIT PAS être enregistré pour DCCP, car ce nom d'accès implique un transport fiable. Par exemple, on déconseille l'enregistrement de tout accès "http" pour DCCP. Cependant, si un tel enregistrement a un sens (c'est-à-dire, s'il y a une intention concrète d'utiliser un tel accès) l'enregistrement DCCP DEVRAIT utiliser le même numéro d'accès que l'enregistrement actuel.
- o Plusieurs enregistrements DCCP pour le même numéro d'accès sont autorisés pour autant que les codes de service des

enregistrements ne se chevauchent pas.

Le présent document enregistre l'accès suivant. (Ceci devrait être considéré comme un enregistrement modèle.)

discard 9/dccp Discard SC:DISC
IETF dccp WG, Eddie Kohler <kohler@cs.ucla.edu>, [RFC4340]

Le service d'élimination *(discard)*, qui accepte les connexions sur l'accès DCCP 9, élimine toutes les données d'application entrantes et n'envoie pas de données en réponse. Donc, l'accès d'élimination de DCCP est analogue à l'accès d'élimination de TCP, et pourrait être utilisé pour vérifier la santé d'une pile DCCP.

20 Remerciements

Merci à Jitendra Padhye pour son aide sur les premières versions de cette spécification.

Merci à Junwen Lai et Arun Venkataramani, qui, en tant que stagiaires au ICIR, ont construit un prototype de mise en œuvre de DCCP. En particulier Lai Junwen a recommandé que le vieux mécanisme de négociation de caractéristique soit abandonné et a co-conçu le mécanisme actuel. Les retours de Arun Venkataramani ont améliorée l'Appendice A.

Nous remercions le personnel et les stagiaires du ICIR, autrefois ACIRI, les membres du groupe de recherche de bout en bout, et les membres du groupe de travail Zone transport de leurs commentaires sur DCCP. Nous remercions tout particulièrement les experts réviseurs de DCCP Greg Minshall, Eric Rescorla, et Magnus Westerlund pour les observations écrites détaillées et le pointage des problèmes, et Rob Austein et Steve Bellovin pour leurs commentaires verbaux et leurs notes écrites. Nous avons aussi à remercier particulièrement Aaron Falk, le président du groupe de travail lors du développement de cette spécification.

Nous remercions également ceux qui ont fourni des commentaires et des suggestions via le BOF DCCP, le groupe de travail, et les listes de diffusion, y compris Damon Lanphear, Patrick McManus, Colin Perkins, Sara Karlberg, Kevin Lai, Bernard Aboba, Youngsoo Choi, Pengfei Di, Dan Duchamp, Lars Eggert, Gorry Fairhurst, Derek Fawcus, David Timothy Fleeman, John Loughney, Ghyslain Pelletier, Paul Hagen Pfeifer, Tom Phelan, Stanislav Shalunov, Somsak Vanit-Anunchai, David Vos, Yufei Wang, et Michael Welzl. En particulier, Colin Perkins a fourni des retours nombreux et détaillés, Michael Welzl a suggéré l'option Somme de contrôle des données, Gorry Fairhurst a fourni de nombreux commentaires sur différentes questions de somme de contrôle, et le modèle de réseau de Petri coloré de Somsak Vanit-Anunchai, Jonathan Billington, et Tul Kongprakaiwoot [VBK05] a découvert plusieurs problèmes avec les échanges de messages.

Appendice A Notes de mise en œuvre du vecteur d'accusé de réception

Le présent Appendice discute des particularités du traitement de l'accusé de réception DCCP dans le contexte d'une mise en œuvre abstraite pour Vecteur Ack. Il est pour information et non normatif.

La première partie de notre mise en œuvre fonctionne à la demi-connexion receveur, et accuse donc réception des paquets de données. Elle génère des options Vecteur Ack. La mise en œuvre a les caractéristiques suivantes :
- o Au plus un octet d'état par paquet acquitté.
- o 0(1) temps pour mettre à jour cet état quand un nouveau paquet arrive (cas normal).
- o Accusés de réception cumulatifs.
- o Élimination rapide de l'état ancien.

La structure de base des données est une mémoire tampon circulaire contenant des informations sur les paquets acquittés. Chaque octet dans cette mémoire tampon contient un état et une longueur de plage ; l'état peut être 0 (paquet reçu), 1 (paquet marqué ECN) ou 3 (paquet pas encore reçu). La mémoire tampon se développe de droite à gauche. La mise en œuvre tient cinq variables, en plus du contenu de la mémoire tampon :
- o "buf_head" et "buf_tail", qui marquent la partie vivante de la mémoire tampon.
  "buf_ackno", le numéro d'accusé de réception du paquet le plus récent reconnu dans la mémoire tampon. Cela correspond au pointeur "tête" *(buf_head)*.
- o "buf_nonce", la somme d'un bit (OU-exclusif, ou parité) des noms occasionnels ECN reçus sur tous les paquets acquittés dans la mémoire tampon avec l'état 0.

On représente comme suit les mémoires tampon d'accusé de réception :

```
+-------------------------------------------------------------+
|S,L|S,L|S,L|S,L|S,L|   |   |   |S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L|
+-------------------------------------------------------------+
              ^                   ^
           buf_tail     buf_head, buf_ackno = A     buf_nonce = E
          <=== buf_head et buf_tail se déplacent dans ce sens <===
```

Chaque "S, L" représente un octet de État/Longueur de plage. On représente ces mémoires tampon en montrant seulement la portion active et on ajoute une annotation montrant le numéro d'accusé de réception pour le dernier octet actif dans la mémoire tampon. Par exemple :

```
  +-----------------------------------------------+
A |S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L|S,L| T    BN[E]
  +-----------------------------------------------+
```

Ici, buf_nonce égale E et buf_ackno égale A.

On va utiliser cette mémoire tampon comme exemple.

```
   +---------------------------+
10 |0,0|3,0|3,0|3,0|0,4|1,0|0,0| 0    BN[1] [Exemple de mémoire tampon]
   +---------------------------+
```

En termes concrets, sa signification est la suivante :
Le paquet 10 a été reçu. (La tête de la mémoire tampon a le numéro de séquence 10, l'état 0, et la longueur de plage 0.)
Les paquets 9, 8, et 7 n'ont pas encore été reçus. (Les trois octets précédant la tête ont l'état 3 et la longueur de plage 0.)
Les paquets 6, 5, 4, 3, et 2 ont été reçus.
Le paquet 1 a été marqué ECN.
Le paquet 0 a été reçu.
La somme d'un bit des noms occasionnels ECN sur les paquets de 10, 6, 5, 4, 3, 2, et 0 est égale à 1.

De plus, la demi-connexion receveur doit garder certaines informations sur les Vecteurs Ack qu'elle a récemment envoyés. Pour chaque paquet envoyé transportant un Vecteur Ack, elle se souvient de quatre variables :

- "ack_seqno", le numéro de séquence utilisé pour le paquet. C'est un numéro de séquence de demi-connexion receveur.
- "ack_ptr", la valeur de buf_head au moment de l'accusé de réception.
- "ack_runlen", la longueur de plage mémorisée dans l'octet des données de mémoire tampon à buf_head au moment de l'accusé de réception..
- "ack_ackno", le numéro d'accusé de réception utilisé pour le paquet. C'est un numéro de séquence de demi-connexion envoyeur. Comme les accusés de réception sont cumulatifs, ce seul numéro spécifie complètement toutes les informations nécessaires sur les paquets acquittés par ce Vecteur Ack.
- "ack_nonce", la somme sur un bit des noms occasionnels ECN pour tous les paquets d'état 0 dans la mémoire tampon de buf_head à ack_ackno, inclus. Initialement, cela équivaut à l'écho du nom occasionnel du Vecteur Ack de l'accusé de réception (ou, si le paquet d'accusé de réception contient plus d'un Vecteur Ack, le OU-exclusif de l'ensemble des Vecteurs Ack des accusés de réception). Cela change lorsque des informations sur les vieux accusés de réception sont supprimées (de sorte que ack_ptr et buf_head divergent) et que les vieux paquets arrivent (de sorte qu'ils changent de l'état 3 ou 1 à l'état 0).

A.1 Arrivée de paquet

Cette section décrit comment la demi-connexion receveur met à jour sa mémoire tampon d'accusé de réception lorsque des paquets arrivent de la demi-connexion envoyeur.

A.1.1 Nouveaux paquets

Quand un paquet avec un numéro de séquence supérieur à buf_ackno arrive, la demi-connexion receveur met à jour buf_head (en le déplaçant vers la gauche de façon appropriée) buf_ackno (qui est mis au numéro de séquence du nouveau paquet) et, éventuellement, buf_nonce (si le paquet est arrivé non marqué avec ECN Nonce 1), en plus de la mémoire tampon elle-même. Par exemple, si le paquet 11 de la demi-connexion envoyeur est arrivé marqué ECN, l'exemple de mémoire tampon ci-dessus entrerait dans ce nouvel état (les changements sont marqués d'une étoile):

```
** +***---------------------------+
11 |1,0|0,0|3,0|3,0|3,0|0,4|1,0|0,0| 0    BN[1]
** +***---------------------------+
```

Si l'état du paquet est égal à l'état à la tête de la mémoire tampon, la demi-connexion receveur peut choisir d'incrémenter sa longueur de plage (jusqu'au maximum). Par exemple, si le paquet 11 de la demi-connexion envoyeur est arrivé sans marquage ECN et avec ECN Nonce 0, l'exemple de mémoire tampon pourraient entrer à la place dans cet état :

```
** +--*-----------------------+
11 |0,1|3,0|3,0|3,0|0,4|1,0|0,0| 0    BN[1]
** +--*-----------------------+
```

Bien sûr, le numéro de séquence du nouveau paquet pourrait ne pas égaler le numéro de séquence attendu. Dans ce cas, la demi-connexion receveur va faire entrer les paquets dans l'état 3. Si plusieurs paquets manquent, la demi-connexion receveur peut préférer entrer plusieurs octets avec longueur de plage 0, plutôt qu'un seul octet avec une longueur de plage supérieure ; cela simplifie la mises à jour des tableaux si l'un des paquets manquants arrive. Par exemple, si le paquet 12 de la demi-connexion envoyeur est arrivé avec ECN Nonce 1, l'exemple de mémoire tampon entrerait cet état :

```
** +*******---------------------------+         *
12 |0,0|3,0|0,1|3,0|3,0|3,0|0,4|1,0|0,0| 0    BN[0]
** +*******---------------------------+         *
```

Bien sûr, la mémoire tampon circulaire peut déborder lorsque la demi-connexion envoyeur envoie des données à un taux très élevé, lorsque les accusés de réception de la demi-connexion receveur n'atteignent pas la demi-connexion envoyeur, ou lorsque la demi-connexion envoyeur oublie d'accuser réception de ces ACK (de sorte que la demi-connexion receveur est incapable de nettoyer l'ancien état). Dans ce cas, la demi-connexion receveur devrait soit compresser la mémoire tampon (en augmentant les longueurs de plage lorsque possible) transférer son état à une plus grande mémoire tampon, ou, en dernier recours, éliminer tous les paquets reçus, sans en traiter aucun, jusqu'à ce que sa mémoire tampon se rétracte à nouveau.

A.1.2 Vieux paquets

Quand un paquet avec le numéro de séquence S ? buf_ackno arrive, la demi-connexion receveur va examiner le tableau à la recherche de l'octet correspondant à S. (Les structures d'indexation pourrait réduire la complexité de cette analyse.) Si S a été auparavant perdu (état 3), et si il a été mémorisé dans un octet avec longueur de plage 0, la demi-connexion receveur peut simplement changer l'état de l'octet. Par exemple, si le paquet 8 de la demi-connexion envoyeur a été reçu avec ECN Nonce 0, l'exemple de mémoire tampon entrerait dans cet état :

```
   +--------*------------------+
10 |0,0|3,0|0,0|3,0|0,4|1,0|0,0| 0    BN[1]
   +--------*------------------+
```

Si S n'a pas été marqué comme perdu, ou si il ne figurait pas dans le tableau, le paquet est probablement un doublon et devrait être ignoré. (L'état marqué ECN du nouveau paquet pourrait différer de l'état dans la mémoire tampon ; le paragraphe 11.4.1 décrit ce qui est alors autorisé.) Si l'octet de mémoire tampon de S a une longueur de plage non nulle, alors la mémoire tampon pourrait avoir besoin d'être remaniée pour faire de la place à un ou deux nouveaux octets.

Les champs ack_nonce peuvent aussi avoir besoin de manipulation lorsque les vieux paquets arrivent. En particulier, lorsque S passe de l'état 3 ou 1 à l'état 0, et que S avait ECN Nonce 1, alors la mise en œuvre devrait sauter la valeur de ack_nonce pour chaque accusé de réception avec ack_ackno? S.

Il est impossible avec cette structure de données de glisser les paquets de l'état 0 à l'état 1, car la mémoire tampon ne stocke pas les noms occasionnels ECN des paquets individuels.

A.2. Envoi des accusés de réception

Chaque fois que la demi-connexion receveur a besoin de générer un accusé de réception, le contenu de la mémoire tampon peut tout simplement être copié dans une ou plusieurs options Vecteur Ack. Les Vecteurs Ack copiés pourraient ne pas être comprimés au maximum ; par exemple, l'exemple de mémoire tampon ci-dessus contient trois octets 3,0 adjacents qui pourraient être combinés en un seul octet 3,2. La demi-connexion receveur pourrait, par conséquent, choisir de compresser la mémoire tampon en place avant l'envoi de l'option, ou de comprimer la mémoire tampon pendant sa copie ; l'une et l'autre opération sont simples.

Chaque accusé de réception envoyé par la demi-connexion receveur DEVRAIT inclure l'état complet de la mémoire tampon. C'est-à-dire, les accusés de réception sont cumulatifs.

Si l'accusé de réception tient dans un Vecteur Ack, l'écho de nom occasionnel de ce Vecteur Ack équivaut tout simplement à buf_nonce. Pour plusieurs Vecteurs Ack, plus de soins sont requis. Les Vecteurs Ack devraient être séparés aux points qui correspondent aux acquittements antérieurs, car les champs ack_nonce mémorisés fournissent suffisamment d'informations pour calculer les échos de noms occasionnel corrects. La mise en œuvre devrait donc acquitter les données au moins une fois par 253 octets d'état de mémoire tampon. (Sinon, il n'y aurait aucun moyen de calculer un écho de nom occasionnel.)

Pour chaque accusé de réception qu'elle envoie, la demi-connexion receveur va ajouter un enregistrement d'accusé de réception. ack_seqno sera égal au numéro de séquence de la demi-connexion receveur utilisé pour le paquet d'accusé de réception ; ack_ptr sera égal à buf_head ; ack_runlen sera égal à la longueur de plage mémorisé dans l'octet buf-head de la mémoire tampon ; ack_ackno sera égal à buf_ackno ; et ack_nonce sera égal à buf_nonce.

A.3. État Clearing

Certains paquets de la demi-connexion envoyeur vont inclure le numéro d'accusé de réception, qui accusent réception des accusés de réception de la demi-connexion receveur. Quand un tel accusé de réception est reçu, la demi-connexion receveur trouve l'enregistrement R de l'accusé de réception avec le ack_seqno approprié puis effectue les opérations suivantes :

- Si la longueur de plage dans l'octet R.ack_ptr de la mémoire tampon est supérieure à R.ack_runlen, elle décrémente alors cette longueur de plage de R.ack_runlen + 1 et règle buf_tail à R.ack_ptr. Sinon, elle règle buf_tail à R.ack_ptr + 1.
- Si R.ack_nonce est 1, cela bascule buf_nonce, et la valeur de ack_nonce pour chaque enregistrement d'accusé de réception ultérieur.
- Elle élimine R et tous les enregistrements précédents d'accusés de réception.

(La demi-connexion receveur peut choisir de garder quelques informations plus anciennes, dans le cas où un paquet considéré comme perdu réapparaîtrait plus tard.) Par exemple, disons que la demi-connexion receveur qui mémorise la mémoire tampon exemple avait déjà envoyé deux accusés de réception :

1. ack_seqno = 59, ack_runlen = 1, ack_ackno = 3, ack_nonce = 1.
2. ack_seqno = 60, ack_runlen = 0, ack_ackno = 10, ack_nonce = 0.

Disons que la demi-connexion receveur a reçu ensuite un paquet DCCP-DataAck avec le numéro d'accusé de réception 59 de la demi-connexion envoyeur. Ceci informe la demi-connexion receveur que la demi-connexion envoyeur a reçu, et traité, toutes les informations dans le paquet 59 de la demi-connexion receveur. Ce paquet accuse réception du paquet 3 de la demi-connexion envoyeur, de sorte que la demi-connexion envoyeur a maintenant reçu les accusés de réception de la demi-connexion receveur pour les paquets 0, 1, 2 et 3. L'exemple de mémoire tampon devrait entrer dans cet état :

```
   +------------------*+ *        *
10 |0,0|3,0|3,0|3,0|0,2| 4     BN[0]
   +------------------*+ *        *
```

La longueur de plage de l'octet de queue a été ajustée, car le paquet 3 était au milieu de cet octet. Comme R.ack_nonce était de 1, le champ buf_nonce a été sauté, comme l'ont été les champs ack_nonce pour des accusés de réception ultérieurs (ici, l'enregistrement Ack 60 de la demi-connexion receveur, non illustré ici, a son ack_nonce retourné à 1). La demi-connexion receveur peut également jeter les informations mémorisées sur l'Ack 59 de la demi-connexion receveur et tout accusé de réception antérieur.

Une mise en œuvre attentive pourrait tenter de s'assurer d'une robustesse raisonnable au réordonnancement. Supposons que l'exemple de mémoire tampon soit comme avant, mais que le paquet de 9 arrive alors, hors séquence. La mémoire tampon entrerait dans cet état :

```
   +----*----------------------+
10 |0,0|0,0|3,0|3,0|0,4|1,0|0,0| 0      BN[1]
   +----*----------------------+
```

Le danger est que la demi-connexion envoyeur pourrait accuser réception de l'accusé de réception précédent de la demi-connexion receveur (avec le numéro de séquence 60) qui dit que le paquet 9 n'a pas été reçu, avant que la demi-connexion receveur ait une chance d'envoyer un nouvel accusé de réception disant que le paquet 9 a en fait été reçu. Par conséquent, lorsque le paquet 9 est arrivé, la demi-connexion receveur pourrait modifier son enregistrement d'accusé de réception comme suit :

1. ack_seqno = 59, ack_ackno = 3, ack_nonce = 1.
2. ack_seqno = 60, ack_ackno = 3, ack_nonce = 1.

C'est-à-dire que Ack 60 est désormais traité comme un doublon de Ack 59. Cela permettra d'éviter que le pointeur de queue passe derrière le paquet 9 jusqu'à ce que la demi-connexion receveur sache que la demi-connexion envoyeur a vu un vecteur Ack indiquant l'arrivée du paquet.

A.4 Traitement des accusés de réception

Lorsque la demi-connexion envoyeur reçoit un accusé de réception, elle s'occupe généralement du nombre de paquets qui ont été abandonnés et/ou marqués ECN. Elle lit simplement cela sur le Vecteur Ack. En outre, elle devrait vérifier l'exactitude du nom occasionnel ECN. (Comme il est décrit au paragraphe 11.4.1, elle veut peut-être garder des informations plus détaillées sur les paquets acquittés au cas où des paquets changeraient d'état entre les accusés de réception, ou si l'application demande si un paquet est arrivé.)

La demi-connexion envoyeur doit également accuser réception des accusés de réception de la demi-connexion receveur afin que celle-ci puisse libérer le vieil état de Vecteur Ack. (Comme les accusés de réception de Vecteur Ack sont fiables, la demi-connexion receveur doit conserver et renvoyer les informations de Vecteur Ack jusqu'à ce qu'elle soit sûre que la demi-connexion envoyeur a reçu cette information.) Un simple algorithme suffit : comme les accusés de réception Vecteur Ack sont cumulatifs, un simple numéro d'accusé de réception dit à la demi-connexion receveur combien d'informations d'accusé de réception sont arrivées. En supposant que la demi-connexion receveur n'envoie pas de données, la demi-connexion envoyeur peut s'assurer qu'au moins une fois par temps d'aller-retour, elle envoie un paquet DCCP-DataAck accusant réception du dernier paquet DCCP-Ack qu'elle a reçu. Bien sûr, la demi-connexion envoyeur a seulement besoin d'accuser réception des accusés de réception de la demi-connexion receveur si elle envoie également des données. Si la demi-connexion envoyeur n'envoie pas de données, alors l'état de Vecteur Ack de la demi-connexion receveur est stable, et il n'est pas nécessaire de le rétrécir. La demi-connexion envoyeur doit regarder les abandons et les marques ECN sur les paquets DCCP-Ack reçus afin qu'elle puisse ajuster le taux d'envoi des accusés de réception de la demi-connexion receveur en réponse à l’encombrement, par exemple, avec Ack Ratio.

Si l'autre demi-connexion n'est pas au repos -- c'est-à-dire, si la demi-connexion receveur envoie des données à la demi-connexion envoyeur, éventuellement en utilisant un autre CCID -- alors les accusés de réception sur cette demi-connexion sont suffisants pour que la demi-connexion receveur libère son état.

Appendice B. Motivation du concept de somme de contrôle partielle

De grandes discussions ont eu lieu sur l'utilité de permettre à un envoyeur DCCP de restreindre la somme de contrôle de sorte qu'elle ne couvre pas le paquet complet. Cette section tente de décrire une partie des raisons des détails spécifiques de la conception de DCCP.

La plupart des applications que nous envisageons à l'aide de DCCP sont résistantes à certains degrés de perte de données, ou elles ont généralement choisi un transport fiable. Certaines de ces applications peuvent également être résilientes à la corruption des données -- certaines charges utiles audio, par exemple. Ces applications résilientes pourraient préférer recevoir des données corrompues que d'avoir un abandon des paquets DCCP corrompus. C'est en particulier en raison du contrôle d’encombrement : DCCP ne peut pas faire la différence entre les paquets éliminés à cause de la corruption et les paquets perdus en raison de l’encombrement, et donc il doit réduire le taux de transmission en conséquence. Cette réponse peut provoquer la réception par la connexion de moins de bande passante qu'il n'est dû ; la corruption dans certaines technologies de réseaux est indépendante de, ou du moins, pas toujours corrélée à, l'encombrement. Par conséquent, les paquets corrompus n'ont pas besoin de causer une aussi forte réduction du taux de transmission que le dicterait la réponse d’encombrement (tant que l'en-tête DCCP et les options ne sont pas corrompus).

Ainsi DCCP permet que la somme de contrôle couvre tout le paquet, juste l'en-tête DCCP, ou à la fois l'en-tête DCCP et un certain nombre d'octets des données d'application. Si l'application ne peut tolérer aucune corruption des données, alors la somme de contrôle doit couvrir l'ensemble du paquet. Si l'application préfère tolérer une certaine corruption plutôt que d'avoir le paquet éliminé, alors elle peut régler la somme de contrôle à couvrir une partie seulement du paquet (mais toujours l'en-tête DCCP). En outre, si l'application souhaite découpler la somme de contrôle de l'en-tête DCCP de la somme de contrôle des données d'application, elle peut le faire en incluant l'option Somme de contrôle des données. Cela permettra à DCCP d'écarter les données d'applications corrompues sans prendre la corruption pour l’encombrement du réseau.

Ainsi, du point de vue de l'application, les sommes de contrôle partielles semblent être une caractéristique désirable. Toutefois, l'utilité des sommes de contrôle partielles dépend de la livraison de paquets partiellement corrompus au receveur.

Si le CRC de couche de liaison rejette toujours les paquets corrompus, cela n'arrivera alors pas, et ainsi l'utilité des sommes de contrôle partielles sera limitée à la corruption qui s'est produite dans les routeurs et les autres endroits non couverts par les CRC de liaison. Il ne semble pas qu'il y ait de consensus sur la probabilité que les liaisons des futurs réseaux qui subissent une corruption significative ne couvrent pas l'ensemble du paquet avec un seul fort CRC. DCCP permet d'adapter de telles liaisons aux applications, mais il est difficile de prédire si ce sera convaincant pour les technologies des futures liaisons.

En outre, les sommes de contrôle partielles ne co-existent pas bien avec les mécanismes d'authentification de niveau IP tels que IPsec AH, qui couvrent l'ensemble du paquet avec un hachage cryptographique. Donc, si des mécanismes d'authentification cryptographiques doivent nécessairement co-exister avec des sommes de contrôle partielles, l'authentification doit être effectuée dans les données d'application. Un mode d'utilisation possible pourrait être similaire à celui de RTP sécurisé. Cependant, une telle authentification de "niveau application" ne protège pas la négociation d'option DCCP et l'automate à états contre les paquets falsifiés. Une alternative serait d'utiliser IPsec ESP, et d'utiliser le chiffrement pour protéger les en-têtes DCCP contre les attaques, tout en utilisant la vérification de la validité de l'en-tête DCCP pour authentifier que l'en-tête vient de quelqu'un qui possédait la clef correcte. Bien que ce soit résistant à la répétition (en raison du numéro de séquence DCCP) ce n'est pas en soi résistant à certaines formes d'attaques par interposition parce que les données d'application ne sont pas étroitement couplées à l'en-tête du paquet. Donc, une authentification de niveau application doit probablement être couplée avec IPsec ESP ou un mécanisme similaire pour fournir une solution de sécurité assez complète. Le surcoût d'une telle solution pourrait être inacceptable pour certaines applications qui souhaiteraient autrement utiliser la somme de contrôle partielle.

Cependant, les auteurs estiment que les sommes de contrôles DCCP partielles permettent potentiellement certains usages futurs qui seraient difficiles autrement. Comme le coût et la complexité de leur prise en charge sont petits, il semble utile de les inclure pour le moment. Il reste à voir si elles sont utiles dans la pratique.

Références normatives

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.

[RFC1191] J. Mogul et S. Deering, "Découverte de la MTU de chemin", novembre 1990.

[RFC2019] M. Crawford, "Transmission de paquets IPv6 sur FDDI", octobre 1996. (*Obsolète, voir* RFC2467) *(P.S.)*

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la* RFC5226*)*

[RFC2460] S. Deering et R. Hinden, "Spécification du protocole Internet, version 6 (IPv6)", décembre 1998. *(MàJ par* RFC5095*, D.S)*

[RFC3168] K. Ramakrishnan et autres, "Ajout de la notification explicite d'encombrement (ECN) à IP", septembre 2001. *(P.S.)*

[RFC3309] J. Stone, R. Stewart, D. Otis, "Changement de somme de contrôle du protocole de transmission de commandes de flux (SCTP)". septembre 2002. (*Obsolète, voir* RFC4960) *(P.S.)*

[RFC3692] T. Narten, "L'allocation de numéros expérimentaux et d'essai est considérée comme utile", janvier 2004. (BCP0082)

[RFC3775] D. Johnson, C. Perkins, J. Arkko, "Prise en charge de la mobilité dans IPv6", juin 2004. *(P.S.)*

[RFC3828] L-A. Larzon et autres, "Protocole léger de datagramme d'utilisateur (UDP-Lite)", juillet 2004. *(P.S.)*

Références pour information

[B98] Bellovin, SM, "La cryptographie et l'Internet", CRYPTO '98 (LNCS 1462), pp 46-55, août 1988.

[BB01] Bellovin, S.M. M. et Blaze, "Modes de chiffrement des opérations pour l'Internet", 2ème atelier sur le mode de fonctionnement du NIST, août 2001.

[M85] Morris, RT, "Une faiblesse dans le logiciel TCP/IP 4.2BSD Unix", Computer Science Technical Report 117, AT & T Bell Laboratories, Murray Hill, NJ, février 1985.

[RFC0792] J. Postel, "Protocole du message de contrôle Internet – Spécification du protocole du programme Internet DARPA", STD 5, septembre 1981.

[RFC1812] F. Baker, "Exigences pour les routeurs IP version 4", juin 1995. *(Mise à jour par la* RFC 2644*)*

[RFC1948] S. Bellovin, "Se défendre contre les attaques de numéro de séquence"**,** mai 1996. *(Information)*

[RFC1982] R. Elz, R. Bush, "Arithmétique des numéros de série", août 1996. (*MàJ* RFC1034, RFC1035) *(P.S.)*

[RFC2018] M. Mathis et autres, "Options d'accusé de réception sélectif sur TCP", octobre 1996. *(P.S.)*

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2463] A. Conta, S. Deering, "Protocole de message de contrôle Internet (ICMPv6) pour le protocole Internet v.6 (IPv6)", décembre 1998. (*Obsolète, voir* RFC4443) *(D.S.)*

[RFC2581] M. Alman, V. Paxson et W. Stevens, "Contrôle d'encombrement avec TCP", avril 1999. *(Obsolète, voir* RFC5681)

[RFC2960] R. Stewart et autres, "Protocole de transmission de commandes de flux", octobre 2000. (*Obsolète, voir* RFC4960) (*MàJ par* RFC3309) *(P.S.)*

[RFC3124] H. Balakrishnan et S Seshan, "Le gestionnaire d'encombrement", juin 2001.

[RFC3360] S. Floyd, "Réinitialisations inappropriées de TCP considérées comme dommageables", août 2002. (BCP0060)

[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.)*

[RFC3540] N. Spring, D. Wetherall, D. Ely, "Signalisation de notification robuste d'encombrement explicite (ECN) avec des noms occasionnels", juin 2003. *(Expérimentale)*

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications en temps réel", STD 64, juillet 2003.

[RFC3611] T. Friedman, R. Caceres et A. Clark, éditeurs, "Rapports étendus du protocole de contrôle de RTP (RTCP XR)", novembre 2003. *(P.S.)*

[RFC3711] M. Baugher et autres, "Protocole de transport sécurisé en temps réel (SRTP)", mars 2004.

[RFC3819] P. Karn et autres, "Conseils aux concepteurs de sous-réseaux Internet", juillet 2004. (BCP0089)

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005. (*Remplace* RFC1750) (BCP0106)

[RFC4341] S. Floyd, E. Kohler, "Profil d'identifiant 2 de protocole de contrôle d'encombrement de datagrammes (DCCP) : Contrôle d'encombrement de style TCP", mars 2006. *(P.S.)*

[RFC4342] S. Floyd et autres, "Profil d'identifiant 3 de protocole de contrôle d'encombrement de datagrammes (DCCP) : Contrôle en douceur de débit TCP (TFRC)", mars 2006. (*MàJ par* RFC5348) *(P.S.)*

[RFC4821] M. Mathis, J. Heffner, "Découverte de la MTU de chemin de couche de mise en paquet", mars 2007. *(P.S.)*

[SHHP00] Spatscheck O., Hansen JS, Hartman JH, et LL Peterson, "Optimisation des performances d'émetteur TCP", IEEE / ACM Transactions on Networking 8 (2) :146-157, avril 2000.

[SYNCOOKIES] Bernstein, D.J., "SYN Cookies", http://cr.yp.to/syncookies.html , mars 2006.

[VBK05] Vanit-Anunchai S., Billington J., and T. Kongprakaiwoot, "Discovering Chatter and Incompleteness in the Datagram Congestion Control Protocol", FORTE 2005, pp 143-158, octobre 2005.

Adresses des auteurs

Eddie Kohler
4531C Boelter salle
UCLA Département Informatique
Los Angeles, CA 90095
USA
mél : kohler@cs.ucla.edu

Mark Handley
Département des sciences informatiques
University College London
Gower Street
Londres WC1E 6BT. UK
mél : M.Handley@cs.ucl.ac.uk

Sally Floyd
ICSI Center for Internet Research
1947 Centre Street, Suite 600
Berkeley, CA 94704
USA
mél : floyd@icir.org

Déclaration complète de droits de reproduction

Remerciement
Le financement des fonctions d'édition des RFC est actuellement fourni par l'activité de soutien administratif (IASA) de l'IETF.