Groupe de travail Réseau
Request for Comments : RFC-908
Traduction Claude Brière de L'Isle

David Velten, Robert Hinden, Jack Sax
BBN Communications Corporation
juillet 1984

Protocole de données fiable

Statut de ce mémoire
La présente RFC spécifie un protocole proposé pour la communauté ARPA Internet, et appelle à des discussions et suggestions pour son amélioration. La distribution du présent mémoire n'est soumise à aucune restriction.

Table des matières

1. Introduction

Le protocole de données fiables (RDP, *Reliable Data Protocol*) a été conçu pour fournir un service fiable de transport de données aux applications fondées sur le paquet telles que le chargement à distance et le déboguage. Le protocole est destiné à être de mise en œuvre simple mais quand même efficace dans les environnements où il peut y avoir de longs délais de transmission et des pertes ou une livraison non séquentielle des segments de message.

Bien que le présent protocole ait été conçu pour des applications telles que le chargement à distance et le déboguage, il peut convenir pour d'autres applications qui exigent des services de message fiables, tels que la messagerie électronique, le transfert de fichiers, le traitement de transactions, etc.

Certains des concepts utilisés proviennent de sources diverses. Les auteurs souhaitent en remercier Eric Rosen, Rob Gurwitz, Jack Haverty, et noter l'emprunt de matériel adapté de la "RFC-793, "Protocole de contrôle de transmission", édité par Jon Postel. Merci à John Linn pour l'algorithme de somme de contrôle.

2. Description générale

2.1 Motivation

RDP est un protocole de transport conçu pour une prise en charge efficace du transfert en gros de données pour des applications de surveillance et contrôle d'hôte telles que le chargement/déchargement et le déboguage à distance. Il essaye de ne fournir que les services nécessaires, afin d'être d'un fonctionnement efficace et de petite taille. Avant de passer à la conception du protocole, il a été nécessaire de considérer quel ensemble minimum de fonctions de transport satisferait aux exigences des applications prévues.

Voici une liste des exigences d'un tel protocole de transport :

- La livraison fiable des paquets est requise. Lors du chargement ou déchargement d'une image mémoire, il est nécessaire que tous les segments de mémoire soient livrés. Un 'trou' laissé dans l'image mémoire n'est pas acceptable. Cependant, l'environnement internet est sujet aux pertes et les paquets peuvent y être endommagés ou perdus. De sorte qu'un mécanisme d'accusés de réception positifs et de retransmission est un composant nécessaire du protocole.
- Comme le chargement et déchargement d'images mémoire sur l'internet implique le transfert en vrac de grandes quantités de données sur un réseau sujet aux pertes avec de longs retards potentiels, il est nécessaire que le protocole déplace efficacement les données. En particulier, les retransmissions inutiles de segments devraient être évitées. Si un seul segment a été perdu mais si les segments suivants ont été correctement reçus, le protocole ne devrait pas exiger la retransmission de tous les segments.
- Le chargement et le déchargement sont des applications qui n'exigent pas nécessairement la livraison ordonnée des segments, pour autant que tous les segments soient finalement livrés. De sorte que le protocole n'a pas besoin de forcer la livraison séquencée. Pour ces types d'applications, les segments peuvent être livrés dans l'ordre dans lequel ils arrivent.
- Cependant, certaines applications peuvent avoir besoin de connaître si un bloc particulier de données a été livré avant d'envoyer le suivant. Par exemple, un débogueur va vouloir savoir si une commande qui insère un point de coupure dans l'image mémoire d'un hôte a été livrée avant d'envoyer une commande "continuer". Si ces segments sont arrivés dans le désordre, le résultat attendu ne sera pas atteint. Le protocole devrait permettre à un usager de spécifier facultativement qu'une connexion doit livrer les segments dans l'ordre des numéros de séquence.
- Les applications de chargement/déchargement et de déboguage qui sont bien définies se prêtent elles mêmes à une mise en paquet facile des données transférées. Elles n'exigent pas un mécanisme complexe de transfert de flux d'octets.

Afin de combiner les exigences du transfert de données en vrac et de la livraison fiable, il est nécessaire de concevoir un protocole fondé sur la connexion qui utilise la prise de contact à trois phases pour synchroniser les numéros de séquence. Le protocole semble se rapprocher de TCP pour la complexité, donc, pourquoi ne pas avoir en fait choisi TCP ? La réponse est que TCP présente certains inconvénients pour ces applications. En particulier :

- TCP est orienté vers un environnement plus général, prenant en charge le transfert d'un flux d'octets entre deux parties d'une communication. TCP est mieux adapté à un environnement dans lequel il n'y a pas de démarcation évidente des données dans un échange de communications. La plus grande partie de la difficulté du développement d'une mise en œuvre de TCP découle de la complexité de la prise en charge de ce transfert général de flux d'octets, et donc une partie significative de la complexité peut être évitée en utilisant un autre protocole. Cette considération n'est pas seulement de mise en œuvre, mais aussi d'efficacité.
- Comme TCP ne permet pas d'accuser réception d'un octet avant que tous les octets antérieurs aient été acquittés, il force souvent à des retransmissions inutiles de données. Donc, il ne satisfait pas à une autre des exigences énoncées ci-dessus.
- TCP fournit une livraison en séquence des données à l'application. Si l'application n'exige pas une telle livraison en séquence, de grandes quantités de ressources sont gaspillées pour la fournir. Par exemple, des mémoires tampon peuvent être utilisées à garder des données jusqu'à ce qu'arrive un segment qui porte un numéro de séquence antérieur. Le protocole ne devrait pas forcer l'application à subir son désir de séquençage de segment.

RDP prend en charge en ensemble de fonctions beaucoup plus simple que TCP. Les schémas de contrôle de flux, de mise en mémoire tampon, et de gestion de connexion pour RDP sont considérablement plus simples et moins complexes. Le but est un protocole qui puisse être facilement et efficacement mis en œuvre et qui serve à toute une gamme d'applications.

Les fonctions de RDP peuvent aussi être réparties en sous-ensembles pour mieux réduire la taille d'une mise en œuvre particulière. Par exemple, un processeur cible qui requiert un téléchargement à partir d'un autre hôte pourrait mettre en œuvre un module RDP qui ne prenne en charge que la fonction passive Open et une seule connexion. Le module pourrait aussi choisir de ne pas mettre en œuvre les accusés de réceptions décalés.

2.2 Relation aux autres protocoles

RDP est un protocole transport qui est à l'aise dans l'environnement internet de couches de protocoles. La Figure 1 illustre la place de RDP dans la hiérarchie des protocoles :

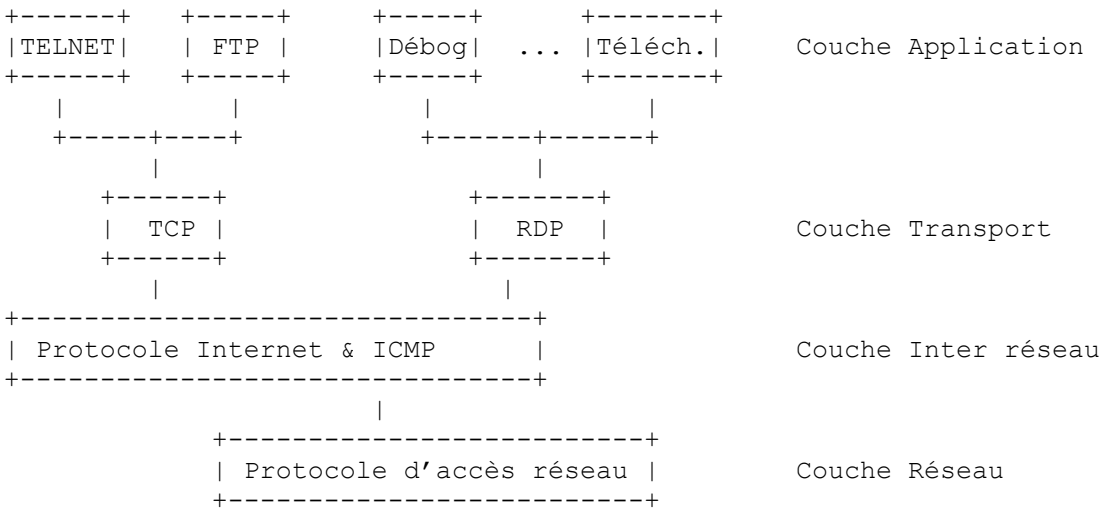

Figure 1 : Relation aux autres protocoles

RDP fournit à la couche application un service fiable de transport de message. L'interface entre les usagers et RDP transfère les données en unités de messages. Lorsque il est mis en œuvre dans l'environnement Internet, RDP est mis en couche par dessus le protocole IP *(Internet Protocol)* qui fournit un service non fiable de datagrammes à RDP. Les données sont passées à travers l'interface RDP/IP sous la forme de segments. RDP utilise les primitives d'interface IP standard pour envoyer et recevoir les segments RDP comme des datagrammes IP. Au niveau de l'Internet, IP échange des datagrammes IP avec la couche réseau. Un paquet internet peut contenir un datagramme entier ou un fragment de datagramme.

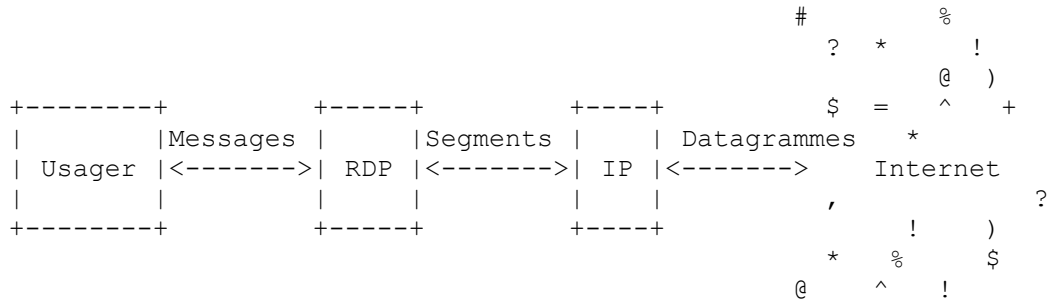

Figure 2 : Forme de l'échange de données entre les couches

Si les services inter réseaux ne sont pas requis, il devrait être possible d'utiliser le RDP sans la couche IP. Tant que le protocole encapsulant fournit à RDP les informations nécessaires comme l'adressage et le démultiplexage de protocole, il devrait être possible de faire fonctionner RDP en couche sur divers protocoles différents.

3. Fonctionnement du protocole

3.1 Objectifs de service du protocole

Les buts du protocole RDP sont les suivants :

- RDP assurera un canal de communications bidirectionnel entre les deux accès de chaque connexion de transport.
- RDP tentera une livraison fiable de tous les messages d'usager et fera rapport des défaillances à l'usager si il ne peut pas livrer un message. RDP étend le service de datagrammes d'IP pour y inclure la livraison fiable.
- RDP va tenter de détecter et éliminer tous les segments endommagés et dupliqués. Il va utiliser une somme de contrôle et un numéro de séquence dans chaque en-tête de segment pour réaliser cet objectif.
- RDP va facultativement fournir la livraison en séquence des segments. La livraison en séquence des segments doit être spécifiée lors de l'établissement de la connexion.

- RDP va accuser réception des segments reçus hors séquence, lorsque ils arrivent. Cela va libérer des ressources du côté envoyeur.

3.2 Gestion de la connexion RDP

RDP est un protocole en mode connexion dans lequel chaque connexion agit comme un canal de communication bidirectionnel entre les deux processeurs. Les segments provenant d'un envoyeur sont dirigés sur un accès chez l'hôte de destination. Les identifiants de deux octets d'accès de source et de destination de l'en-tête RDP sont utilisés en conjonction avec les adresses de source et destination du réseau pour identifier de façon univoque chaque connexion.

3.2.1 Ouverture d'une connexion

Les connexions sont ouvertes en produisant la demande Open, qui peut être active ou passive. Une demande Open passive met RDP dans l'état Listen, durant lequel elle attend passivement une demande d'ouverture d'une connexion provenant d'un site distant. La demande Open active tente d'établir une connexion avec un accès spécifié d'un site distant.

La demande Open active exige que soient spécifiés un accès distant et une adresse d'hôte spécifiques avec la demande. La demande Open passive peut facultativement spécifier un accès distant et une adresse réseau spécifiques, ou elle peut spécifier qu'une demande Open soit acceptée de la part de n'importe qui. Si un accès distant et une adresse d'hôte spécifiques ont été spécifiés, une demande arrivant pour ouvrir une connexion doit exactement correspondre à l'accès distant et adresse spécifiés.

3.2.2 Accès

La gamme de numéros d'accès valide va de 1 à 255 (décimal). Il y a deux types d'accès : les accès "bien connus" et les accès "allouables". Les accès bien connus ont des numéros dans la gamme de 1 à 63 (décimal) et les accès allouables reçoivent des numéros dans la gamme de 64 à 255.

L'usager, quand il produit une demande Open active, doit spécifier à la fois l'hôte distant et l'accès, et peut facultativement spécifier l'accès local. Si l'accès local n'était pas spécifié, RDP va choisir un accès non utilisé dans la gamme des accès allouables. Lorsque il produit une demande Open passive, l'usager doit spécifier le numéro d'accès local. Généralement, dans ce cas, l'accès local sera un accès bien connu.

3.2.3 États de connexion

Une connexion RDP va progresser à travers une série d'états durant sa durée de vie. Les états sont illustrés à la Figure 3 et sont décrits individuellement ci-dessous. Dans la Figure 3, les boîtes représentent les états de l'automate à états finis RDP et les arcs représentent les changements d'état. Chaque arc est marqué avec l'événement qui cause le changement d'état et son résultat.

CLOSED
L'état CLOSED existe lorsque aucune connexion n'existe et qu'il n'y a pas d'enregistrement de connexion d'alloué.

LISTEN
L'état LISTEN est occupé après le traitement d'une demande Open passive. Un enregistrement de connexion est alloué et RDP attend une demande active de la part d'un site distant pour établir une connexion.

SYN-SENT
L'état SYN-SENT est occupé après le traitement d'une demande Open active. Un enregistrement de connexion est alloué, un numéro de séquence initial est généré, et un segment SYN est envoyé au site distant. RDP attend alors dans l'état SYN-SENT un accusé de réception à sa demande Open.

SYN-RCVD
L'état SYN-RCVD peut être atteint soit à partir de l'état LISTEN, soit à partir de l'état SYN-SENT. SYN-RCVD est atteint à partir de l'état LISTEN lorsque un segment SYN demandant une connexion est reçu d'un hôte distant. En réponse, le RDP local génère un numéro de séquence initial pour son côté de la connexion, et envoie ensuite le numéro de séquence et un accusé de réception du segment SYN au côté distant. Il attend ensuite un accusé de réception.

L'état SYN-RCVD est atteint à partir de l'état SYN-SENT lorsque un segment SYN est reçu de l'hôte distant sans que l'accompagne un accusé de réception du segment SYN envoyé à cet hôte distant par le RDP local. Cette situation est causée par des tentatives simultanées d'ouverture d'une connexion, les segments SYN se passant l'un l'autre en transit. L'action est de répéter le segment SYN avec le même numéro de séquence, mais incluant maintenant un ACK du segment SYN de l'hôte

distant pour indiquer l'acceptation de la demande Open.

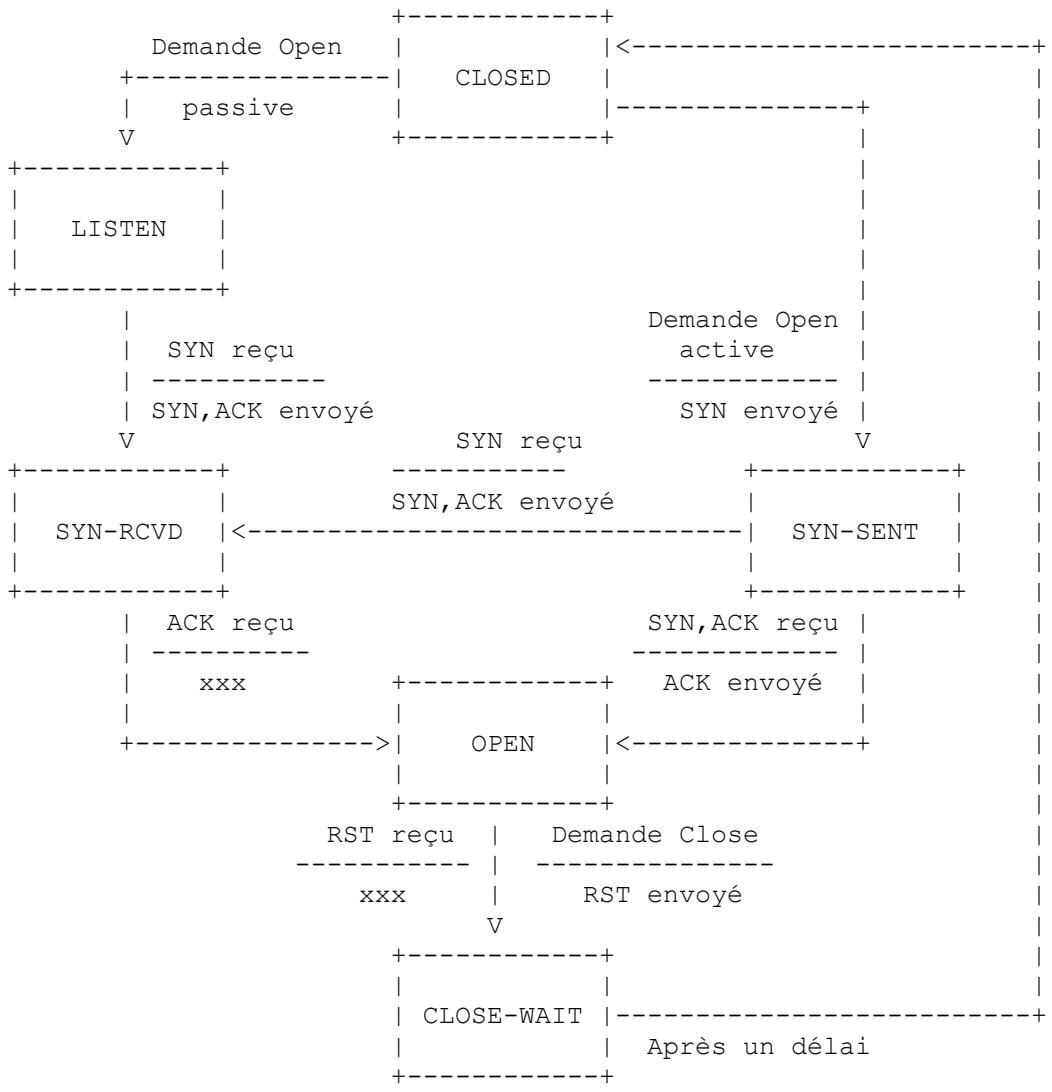

Figure 3 : Diagramme des états de connexion RDP

OPEN
L'état OPEN existe lorsque une connexion a été établie par l'échange réussi des informations d'état entre les deux côtés de la connexion. Chaque côté a échangé et reçu de telles données comme numéro de séquence initial, taille maximum de segment, et nombre maximum de segments non acquittés qui peuvent être en cours. Dans l'état Open des données peuvent être envoyées entre les deux parties de la connexion.

CLOSE-WAIT
L'état CLOSE-WAIT est obtenu à partir soit d'une demande Close, soit de la réception d'un segment RST de la part du site distant. RDP a envoyé un segment RST et attend pendant un certain délai que se termine l'activité sur la connexion.

3.2.4 Enregistrement de connexion

Les variables qui définissent l'état d'une connexion sont mémorisées dans un enregistrement de connexion entretenu pour chaque connexion. Certaines des variables qui seraient mémorisées dans un enregistrement normal de connexion RDP sont décrites ci après. Ceci n'est pas destiné à la spécification d'une mise en œuvre et n'est pas une description complète. L'objet de la désignation et de la description de certains des champs d'un enregistrement de connexion est de simplifier la description du fonctionnement du protocole RDP, en particulier le traitement d'événements.

Voici les champs d'enregistrement de connexion et leur description :
STATE
État en cours de la connexion. Les valeurs légales sont OPEN, LISTEN, CLOSED, SYN-SENT, SYN-RCVD, et CLOSE-WAIT.

Variables de numéro de séquence envoyé :

SND.NXT
Numéro de séquence du prochain segment à envoyer.

SND.UNA
Numéro de séquence du plus ancien segment non acquitté.

SND.MAX
Nombre maximum de segments en cours (non acquittés) qui puisse être envoyé. L'envoyeur ne devrait pas envoyer plus que ce nombre de segments sans obtenir un accusé de réception.

SND.ISS
Numéro de séquence initial envoyé. C'est le numéro de séquence qui a été envoyé dans le segment SYN.

Variables de numéro de séquence reçu :

RCV.CUR
Numéro de séquence du dernier segment reçu correctement et en séquence.

RCV.MAX
Nombre maximum de segments qui peuvent être mis en mémoire tampon pour cette connexion.

RCV.IRS
Numéro de séquence initial reçu. C'est le numéro de séquence du segment SYN qui a établi cette connexion.

RCVDSEQNO[n]
Matrice des numéros de séquence des segments qui ont été reçus et acquittés hors séquence.

Autres variables :

CLOSEWAIT
Temporisateur utilisé pour borner l'état CLOSE-WAIT.

SBUF.MAX
Plus grand segment possible (en octets) qui peut être envoyé légalement. Cette variable est spécifiée par l'hôte étranger dans le segment SYN durant l'établissement de la connexion.

RBUF.MAX
Plus grand segment possible (en octets) qui peut être reçu. Cette variable est spécifiée par l'usager lorsque la connexion est ouverte. La variable est envoyée à l'hôte étranger dans le segment SYN.

Variables du segment en cours :

SEG.SEQ
Numéro de séquence du segment en cours de traitement.

SEG.ACK
Numéro de séquence d'accusé de réception dans le segment actuellement traité.

SEG.MAX
Nombre maximum de segments en cours que le receveur accepte de détenir, comme spécifié dans le segment SYN qui a établi la connexion.

SEG.BMAX
Taille maximum de segment (en octets) acceptée par l'hôte étranger sur une connexion, comme spécifié dans le segment SYN qui a établi la connexion.

3.2.5 Fermeture d'une connexion

La fermeture d'une connexion peut être initiée par une demande Close à partir du traitement d'usager ou par réception d'un

segment RST provenant de l'autre extrémité de la connexion. Dans le cas de la demande Close, RDP va envoyer un segment RST à l'autre côté de la connexion puis entrer dans l'état CLOSE-WAIT pendant un certain temps. Dans l'état CLOSE-WAIT, RDP va éliminer les segments reçus de l'autre côté de la connexion. À l'expiration de la période de temporisation, l'enregistrement de connexion est retiré et la connexion cesse d'exister. Cette simple facilité de fermeture de connexion exige que les usagers déterminent que toutes les données aient été fiablement livrées avant de demander une clôture de la connexion.

3.2.6 Détection d'une connexion semi ouverte

Si un côté d'une connexion tombe en panne, la connexion peut rester avec l'autre côté toujours actif. Cette situation est appelée une connexion semi ouverte. Dans de nombreux cas, le RDP actif va finalement détecter la connexion semi ouverte et réamorcer. Des exemples de récupération de connexions semi ouvertes sont fournis aux paragraphes 5.7 et 5.8. La récupération est normalement réalisée par une activité de l'utilisateur ou par les tentatives de l'hôte en panne pour rétablir la connexion.

Cependant, il y a des cas où la récupération n'est pas possible sans action du RDP lui-même. Par exemple, si tous les blocs de connexion sont utilisés, les tentatives de rétablissement d'une connexion interrompue seront rejetées. Dans ce cas, le RDP peut tenter de libérer des ressources en vérifiant que les connexions sont pleinement ouvertes. Il fait cela en envoyant un segment NUL à chacun des autres RDP. Un accusé de réception indique que la connexion est toujours ouverte et valide.

Pour minimiser la surcharge du réseau, la vérification des connexions ne devrait être faite que lorsque nécessaire pour empêcher une situation de blocage. Seules les connexions inactives devraient être vérifiées. Une connexion inactive se définit comme une connexion qui n'a pas de segment non acquitté en cours, n'a pas de segment dans les files d'attente d'entrée ou de sortie de l'usager, et n'a pas eu de trafic depuis un certain temps.

3.3 Communication des données

Les données s'écoulent à travers une connexion RDP sous la forme de segments. Chaque message d'utilisateur soumis avec une demande Send est mis en paquet pour le transport comme un seul segment RDP. Chaque segment RDP est mis en paquet comme en-tête RDP et un ou plusieurs octets de données. RDP ne va pas tenter de fragmenter un grand message d'utilisateur en plus petits segments et réassembler le message à l'extrémité de réception. Cela diffère d'un protocole en flux d'octets comme TCP qui prend en charge le transfert de flux de données de longueur indéterminée entre les accès, mettant les données en mémoire tampon jusqu'à ce qu'elles soient demandées par le receveur.

Au niveau RDP, les segments sortants, au fur et à mesure de leur création, sont mis en file d'attente comme entrée de la couche IP. Chaque segment est détenu par le RDP envoyeur jusqu'à ce qu'il en soit accusé réception par l'hôte étranger. Les segments entrants sont mis en file d'attente comme entrées du traitement de l'utilisateur à travers l'interface d'utilisateur. Les segments sont acquittés lorsque ils ont été acceptés par le RDP receveur.

L'extrémité receveuse de chaque connexion spécifie la taille maximum de segment qu'elle veut accepter. Toute tentative de l'envoyeur de transmettre un plus grand segment est une erreur. Si RDP détermine qu'une mémoire tampon soumise avec une demande Send excède la taille maximum de segment permise sur la connexion, RDP va retourner une erreur à l'utilisateur. De plus, RDP va interrompre une connexion avec un segment RST si un segment entrant contient plus de données que la taille maximum de segment acceptable. Aucune tentative ne sera faite pour récupérer de cette condition d'erreur ou pour la surmonter de quelque autre façon.

Si la livraison des segments en séquence est nécessaire pour une connexion, cette exigence doit être déclarée lors de l'établissement de la connexion. La livraison en séquence est spécifiée lorsque est faite la demande Open. La livraison en séquence des segments sera alors le mode de livraison pour toute la vie de la connexion.

3.4 Communication fiable

RDP met en œuvre un service de message fiable à travers un certain nombre de mécanismes. Ils incluent l'insertion de numéros de séquence et de sommes de contrôle dans les segments, l'accusé de réception positive des segments, et une temporisation des segments manquants et leur retransmission.

3.4.1 Numéros de séquence de segment

Chaque segment qui transporte des données a un numéro de séquence qui l'identifie de façon univoque parmi tous les autres segments sur la même connexion. Le numéro de séquence initial est choisi lorsque la connexion est ouverte et il est choisi en lisant une valeur sur une horloge à accroissement monotone. Chaque fois qu'est transmis un segment contenant des données, le numéro de séquence est incrémenté. Les segments qui ne contiennent pas de données n'incrémentent pas le numéro de

séquence. Cependant, les segments SYN et NUL, qui ne peuvent pas contenir de données, sont des exceptions. Le segment SYN est toujours envoyé avec un numéro de séquence unique, le numéro de séquence initial. Le segment NUL est envoyé avec le prochain numéro de séquence valide.

3.4.2 Sommes de contrôle

Chaque segment RDP contient une somme de contrôle pour permettre au receveur de détecter les segments endommagés. RDP utilise un algorithme de somme de contrôle non linéaire pour calculer une somme de contrôle de 32 bits et opère sur les données en unités de quatre octets (32 bits). La zone qui est couverte par la somme de contrôle inclut à la fois l'en-tête RDP et la zone de données RDP.

Si un segment contient un certain nombre d'octets d'en-tête et de données qui n'est pas un multiple entier de 4 octets, le dernier octet est bourré à droite avec des zéros pour former une quantité de 32 bits pour les besoins du calcul. Les zéros du bourrage ne sont pas transmis au titre du segment. Lors du calcul de la somme de contrôle, le champ de somme de contrôle lui-même est remplacé par des zéros. L'algorithme réel est décrit au paragraphe 4.2.1.

3.4.3 Accusé de réception positif des segments

RDP suppose qu'il dispose seulement d'un service de datagrammes non fiable pour livrer les segments. Pour garantir la livraison des segments dans cet environnement, RDP utilise des accusés de réception positive et la retransmission des segments. Chaque segment qui contient des données, et les segments SYN et NUL, sont acquittés lorsque ils sont correctement reçus et acceptés par l'hôte de destination. Les segments qui ne contiennent qu'un accusé de réception ne sont pas acquittés. Les segments endommagés sont éliminés et ne sont pas acquittés. Les segments sont retransmis lorsque il n'y a pas d'accusé de réception du segment de la part de l'hôte de destination dans les délais prévus.

RDP permet deux types d'accusé de réception. Un accusé de réception cumulatif est utilisé pour accuser réception de tous les segments jusqu'à un numéro de séquence spécifié. Ce type d'accusé de réception peut être envoyé en utilisant des champs de longueur fixée au sein de l'en-tête RDP. Précisément, le fanion de contrôle ACK est établi *(mis à 1)* et le dernier numéro de séquence acquitté est placé dans le champ Numéro d'accusé de réception.

L'accusé de réception étendu ou non cumulatif permet au receveur d'accuser réception des segments hors séquence. Ce type d'accusé de réception est envoyé en utilisant le fanion de contrôle EACK et les champs de longueur variable dans l'en-tête de segment RDP. Les champs de longueur variable de l'en-tête sont utilisés pour conserver les numéros de séquence des segments hors séquence acquittés.

Le type d'accusé de réception utilisé est simplement une fonction de l'ordre dans lequel arrivent les segments. Chaque fois que possible, les segments sont acquittés en utilisant le segment d'accusé de réception cumulatif. Seuls les segments hors séquence sont acquittés en utilisant l'option d'accusé de réception étendu.

Le processus d'utilisateur, lors de l'initiation de la connexion, ne peut pas restreindre le type d'accusé de réception utilisé sur la connexion. Le receveur peut choisir de ne pas mettre en œuvre les accusés de réception des hors séquence. D'un autre côté, l'envoyeur peut choisir d'ignorer les accusés de réception des hors séquence.

3.4.4 Temporisation et retransmission

Des segments peuvent être perdus dans la transmission pour deux raisons : ils peuvent être perdus ou endommagés dû aux effets d'un support de transmission enclin aux pertes; ou ils peuvent être éliminés par le RDP receveur. La politique d'accusé de réception positive exige que le receveur n'accuse réception d'un segment que lorsque le segment a été correctement reçu et accepté.

Pour détecter les segments manquants, le RDP envoyeur doit utiliser un temporisateur de retransmission pour chaque segment transmis. Le temporisateur est réglé à une valeur qui s'approche du délai de transmission du segment dans le réseau. Lorsque un accusé de réception est reçu pour un segment, le temporisateur est annulé pour ce segment. Si le temporisateur arrive à expiration avant qu'un accusé de réception soit reçu pour un segment, ce segment est retransmis et le temporisateur est redémarré.

3.5 Contrôle de flux et gestion de fenêtre

RDP emploie un simple mécanisme de contrôle de flux qui se fonde sur le nombre de segments non acquittés envoyés et sur le nombre maximum admis de segments en cours (non acquittés). Chaque connexion RDP a un ensemble associé de paramètres de contrôle de flux qui inclut le nombre maximum de segments en cours pour chaque côté d'une connexion. Ces paramètres

sont spécifiés lorsque la connexion est ouverte avec la demande Open, chaque côté de la connexion spécifiant ses propres paramètres. Les paramètres sont passés d'un hôte à l'autre dans les segments de connexion initiaux.

Les valeurs spécifiées pour ces paramètres devraient se fonder sur la quantité et la taille des mémoires tampon que le RDP veut allouer à une connexion. Une mise en œuvre de RDP particulière peut régler les paramètres à des valeurs qui sont optimales pour son schéma de mise en mémoire tampon. Une fois que ces paramètres sont réglés, ils restent inchangés tout au long de la vie de la connexion.

RDP emploie le concept de fenêtre de numéros de séquence pour les numéros de séquence acceptables pour le segment. Le bord gauche de la fenêtre est le numéro du dernier numéro de séquence en séquence acquitté plus un. Le bord droit de la fenêtre est égal au bord droit plus deux fois le nombre maximum admis de segments en cours. Le nombre maximum admis de segments en cours est le nombre de segments que le logiciel RDP qui émet est autorisé à envoyer sans recevoir d'accusé de réception.

Les paramètres Contrôle de flux et Gestion de fenêtre sont utilisés comme suit. Le module RDP de l'hôte émetteur envoie des segments jusqu'à ce qu'il atteigne la limite de segments de la connexion spécifiée par le processus de réception. Une fois que cette limite est atteinte, le module RDP émetteur peut seulement envoyer un nouveau segment pour chaque segment acquitté.

Lorsque un segment reçu a un numéro de séquence qui tombe dans la fenêtre d'acceptation, il est acquitté. Si le numéro de séquence est égal au bord gauche (c'est-à-dire, si il est le prochain numéro de séquence attendu) le segment est acquitté avec un accusé de réception cumulatif (ACK). La fenêtre d'acceptation est ajustée par l'ajout de un à la valeur des bords. Si le numéro de séquence est dans la fenêtre d'acceptation mais est hors séquence, il est acquitté avec un accusé de réception non cumulatif (EACK). La fenêtre n'est pas ajustée, mais la réception du segment hors séquence est enregistrée.

Lorsque des segments sont acquittés dans le désordre, le module RDP émetteur ne doit pas transmettre au delà de la fenêtre d'acceptation. Cela pourrait survenir si un segment n'est pas acquitté mais que tous les segments suivants sont reçus et acquittés. Cette condition va fixer le bord gauche de la fenêtre au numéro de séquence du segment non acquitté. Lorsque des segments supplémentaires sont transmis, le prochain segment à envoyer va s'approcher et éventuellement remplacer le bord droit de la fenêtre. À ce point, toute transmission de nouveaux segments va cesser jusqu'à ce que le segment non acquitté reçoive un accusé de réception.

3.6 Interface d'utilisateur

L'interface d'utilisateur de RDP dépend de la mise en œuvre et peut utiliser des appels système, des invocations de fonction ou quelque autre mécanisme. La liste des demandes qui suit n'est pas destinée à suggérer une mise en œuvre particulière.

Demande OPEN
Ouvre une connexion. Les paramètres incluent le type (actif ou passif), l'accès local, l'accès distant, l'adresse de l'hôte distant, la taille maximum de segment, le nombre maximum de segments non acquittés, le mode de livraison (en séquence ou pas). L'identifiant de connexion, incluant tout numéro d'accès alloué, est retourné à l'usager.

Demande SEND
Envoie un message d'utilisateur. Les paramètres incluent l'identifiant de connexion, l'adresse de mémoire tampon et le compte des données.

Demande RECEIVE
Reçoit un message d'utilisateur. Les paramètres incluent l'identifiant de connexion, l'adresse de mémoire tampon et le compte des données.

Demande CLOSE
Clôt une connexion spécifiée. Le seul paramètre est l'identifiant de connexion.

Demande STATUS
Retourne l'état d'une connexion. Les paramètres incluent l'identifiant de connexion et l'adresse de la mémoire tampon d'état.

3.7 Traitement d'événement

Ce paragraphe décrit une séquence possible pour les événements de traitement. Il n'est pas prévu de suggérer une mise en œuvre particulière, mais toute mise en œuvre réelle ne devrait différer de cette description que dans les détails et non dans sa substance. Voici les sortes d'événements qui peuvent survenir :

Demandes d'utilisateur

Open
Send
Receive
Close
Status

Segment arrivant
Un segment arrive

Fins de temporisation
Fin de temporisation de retransmission
Fin de temporisation de Close-Wait

Le traitement d'une demande d'utilisateur se termine toujours par un retour à l'appelant, avec une éventuelle indication d'erreur. Les réponses d'erreur sont données par une chaîne de caractères. Une réponse retardée est aussi possible dans certaines situations et elle est retournée à l'utilisateur par tout événement ou mécanisme de pseudo interruption disponible. Le terme de "signal" est utilisé pour se référer aux réponses retardées.

Le traitement des segments en arrivée suit normalement cette séquence générale : la validité du numéro de séquence est vérifiée et, si il est validé, le segment est mis en file d'attente et traité dans l'ordre des numéros de séquence. Pour tous les événements, sauf spécification d'un changement d'état, RDP reste dans le même état.

3.7.1 Événements de demande d'utilisateur

Les scénarios qui suivent montrent le traitement des événements causés par la production des demandes d'utilisateur :

Demande Open

État CLOSED
Créer un enregistrement de connexion
Si aucun n'est disponible
Retourner "Erreur – ressources insuffisantes"
Fin de condition

Si Open passif
Si l'accès local n'est pas spécifié
Retourner "Erreur – accès local non spécifié"
Fin de condition
Générer SND.ISS
Régler SND.NXT = SND.ISS + 1
SND.UNA = SND.ISS
Remplir SND.MAX, RMAX.BUF à partir des paramètres Open
Régler État = LISTEN
Retour
Fin de condition

Si Open actif
Si l'accès distant n'est pas spécifié
Retourner "Erreur – accès distant non spécifié"
Fin de condition
Générer SND.ISS
Régler SND.NXT = SND.ISS + 1
SND.UNA = SND.ISS
Remplir SND.MAX, RMAX.BUF à partir des paramètres Open
Si l'accès local n'est pas spécifié
Allouer un accès local
Fin de condition
Envoyer <SEQ=SND.ISS><MAX=SND.MAX><MAXBUF=RMAX.BUF><SYN>
Régler État = SYN-SENT
Retourner (accès local, identifiant de connexion)
Fin de condition

État LISTEN
État SYN-SENT
État SYN-RCVD
État OPEN
État CLOSE-WAIT
    Retourner "Erreur - connexion déjà ouverte"

Demande Close

État OPEN
Envoyer <SEQ=SND.NXT><RST>
    Régler État = CLOSE-WAIT
    Démarrer le temporisateur TIMWAIT
    Retour

État LISTEN
Régler État = CLOSED
    Désallouer l'enregistrement de connexion
    Retour

État SYN-RCVD
État SYN-SENT
Envoyer <SEQ=SND.NXT><RST>
    Régler État = CLOSED
    Retour

État CLOSE-WAIT
Retourner "Erreur – fermeture de connexion"

État CLOSE
Retourner "Erreur - connexion non ouverte"

Demande Receive

État OPEN
Si des données sont en instance
     Retourner les données
    autrement
     Retourner l'indication "pas de données"
    Fin de condition

État LISTEN
État SYN-RCVD
État SYN-SENT
Retourner l'indication "pas de données"

État CLOSE
État CLOSE-WAIT
Retourner "Erreur - connexion non ouverte"
Demande Send

État OPEN
Si SND.NXT < SND.UNA + SND.MAX
     Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK><Data>
     Régler SND.NXT = SND.NXT + 1
     Retour
    autrement
     Retourner "Erreur – ressources insuffisantes pour envoyer des données"
    Fin de condition

État LISTEN
État SYN-RCVD
État SYN-SENT

État CLOSE
État CLOSE-WAIT
Retourner "Erreur - connexion non ouverte"

Demande Status

Retourner :
État de connexion (OPEN, LISTEN, etc.)
Nombre de segments non acquittés
Nombre de segments reçus non donnés à l'usager
Taille maximum de segment pour le côté émission de la connexion
Taille maximum de segment pour le côté réception de la connexion

3.7.2 Événements d'arrivée de segment

Les scénarios qui suivent décrivent le traitement de l'événement causé par l'arrivée d'un segment RDP provenant d'un hôte distant. L'hypothèse est faite que le segment a été traité par l'accès local associé à l'enregistrement de connexion.

```
Si État = CLOSED
  Si RST est établi
    Éliminer le segment
    Retour
    Fin de condition
  Si ACK ou NUL est établi
     Envoyer <SEQ=SEG.ACK + 1><RST>
     Éliminer le segment
     Retour
   autrement
     Envoyer <SEQ=0><RST><ACK=RCV.CUR><ACK>
     Éliminer le segment
     Retour
  Fin de condition
 Fin de condition
 Si État = CLOSE-WAIT
  Si RST est établi
     Régler État = CLOSED
     Éliminer le segment
     Annuler le temporisateur TIMWAIT
     Désallouer l'enregistrement de connexion
   autrement
     Éliminer le segment
    Fin de condition
  Retour
Fin de condition
 Si État = LISTEN
  Si RST est établi
    Éliminer le segment
    Retour
    Fin de condition
  Si ACK ou NUL est établi
    Envoyer <SEQ=SEG.ACK + 1><RST>
    Retour
    Fin de condition
  Si SYN est établi
    Régler RCV.CUR = SEG.SEQ
       RCV.IRS = SEG.SEQ
       SND.MAX = SEG.MAX
       SBUF.MAX = SEG.BMAX
    Envoyer <SEQ=SND.ISS><ACK=RCV.CUR><MAX=RCV.MAX><BUFMAX=RBUF.MAX>
        <ACK><SYN>
    Régler État = SYN-RCVD
    Retour
```

```
    Fin de condition
   Si il y a quelque chose d'autre (qu'on ne devrait jamais avoir)
    Éliminer le segment
    Retour
   Fin de condition
  Fin de condition
  Si État = SYN-SENT
   Si ACK est établi
    Si RST n'est pas établi et si SEG.ACK != SND.ISS
     Envoyer <SEQ=SEG.ACK + 1><RST>
    Fin de condition
    Éliminer le segment; Return
   Fin de condition
   Si RST est établi
    Si ACK est établi
     Signaler "Connexion refusée"
     Régler État = CLOSED
     Désallouer l'enregistrement de connexion
    Fin de condition
    Éliminer le segment
    Retour
   Fin de condition
   Si SYN est établi
    Régler RCV.CUR = SEG.SEQ
      RCV.IRS = SEG.SEQ
      SND.MAX = SEG.MAX
      RBUF.MAX = SEG.BMAX
   Si ACK est établi
     Régler SND.UNA = SEG.ACK
     État = OPEN
     Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK>
    autrement
     Régler État = SYN-RCVD
     Envoyer <SEQ=SND.ISS><ACK=RCV.CUR><MAX=RCV.MAX><BUFMAX=RBUF.MAX><SYN><ACK>
   Fin de condition
    Retour
   Fin de condition
   Si il y a quelque chose d'autre
    Éliminer le segment
    Retour
   Fin de condition
  Fin de condition
  Si État = SYN-RCVD
   Si RCV.IRS < SEG.SEQ =< RCV.CUR + (RCV.MAX * 2)
    Numéro de séquence de segment acceptable
   autrement
    Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK>
    Éliminer le segment
    Retour
   Fin de condition
   Si RST est établi
    Si Open passif
      Régler État = LISTEN
    autrement
      Régler État = CLOSED
      Signaler "Connexion refusée"
      Éliminer le segment
      Désallouer l'enregistrement de connexion
   Fin de condition
    Retour
  Fin de condition
   Si SYN est établi
    Envoyer <SEQ=SEG.ACK + 1><RST>
```

```
    Régler État = CLOSED
    Signaler "Rétablissement de connexion"
    Éliminer le segment
    Désallouer l'enregistrement de connexion
    Retour
  Fin de condition
  Si EACK est établi
    Envoyer <SEQ=SEG.ACK + 1><RST>
    Éliminer le segment
    Retour
  Fin de condition
  Si ACK est établi
    Si SEG.ACK = SND.ISS
      Régler État = OPEN
    autrement
      Envoyer <SEQ=SEG.ACK + 1><RST>
      Éliminer le segment
      Retour
    Fin de condition
  autrement
    Éliminer le segment
    Retour
  Fin de condition
  Si il y a des données dans le segment ou si NUL est établi
    Si le segment reçu est en séquence
      Copier les données (si il en est) dans les mémoires tampon d'utilisateur
      Régler RCV.CUR=SEG.SEQ
      Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK>
    autrement
      Si la livraison hors séquence est permise
        Copier les données (si il en est) dans les mémoires tampon d'utilisateur
      Fin de condition
      Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK><EACK><RCVDSEQNO1>...<RCVDSEQNOn>
    Fin de condition
  Fin de condition
Fin de condition
Si État = OPEN
  Si RCV.CUR < SEG.SEQ ≤ RCV.CUR + (RCV.MAX * 2)
    Numéro de séquence de segment acceptable
  autrement
    Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK>
    Éliminer le segment et retour
  Fin de condition
  Si RST est établi
    Régler État = CLOSE-WAIT
    Signaler "Rétablissement de connexion"
    Retour
  Fin de condition
  Si NUL est établi
    Régler RCV.CUR=SEG.SEQ
    Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK>
    Éliminer le segment
    Retour
  Fin de condition
  Si SYN est établi
    Envoyer <SEQ=SEG.ACK + 1><RST>
    Régler État = CLOSED
    Signaler "Rétablissement de connexion"
    Éliminer le segment
    Désallouer l'enregistrement de connexion
    Retour
  Fin de condition
  Si ACK est établi
```

```
    Si SND.UNA ≤ SEG.ACK < SND.NXT
      Régler SND.UNA = SEG.ACK
      Purger les segments acquittés
    Fin de condition
  Fin de condition
  Si EACK est établi
    Purger les segments acquittés
  Fin de condition
  Si il y a des données dans le segment
   Si le segment reçu est en séquence
    Copier les données dans les mémoires tampon d'utilisateur
    Régler RCV.CUR=SEG.SEQ
    Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK>
   autrement
    Si la livraison hors séquence est permise
      Copier les données dans les mémoires tampon d'utilisateur
    Fin de condition
    Envoyer <SEQ=SND.NXT><ACK=RCV.CUR><ACK><EACK><RCVDSEQNO1>...<RCVDSEQNOn>
    Fin de condition
  Fin de condition
 Fin de condition
```

3.7.3 Événements de fin de temporisation

Des événements de fin de temporisation surviennent lorsque un temporisateur arrive à expiration et le signale à RDP. Deux types d'événements de fin de temporisation peuvent se produire, comme décrit ci-dessous :

```
Fins de temporisation de retransmission
    Si la fin de temporisation porte sur le segment en tête de la file d'attente de retransmission
      Renvoyer le segment en tête de la file d'attente
      Redémarrer le temporisateur de retransmission pour le segment
      Remettre en file d'attente le segment dans la file d'attente de retransmission
      Retour
    Fin de condition
Fins de temporisations de CLOSE-WAIT
    Régler État = CLOSED
    Désallouer l'enregistrement de connexion
    Retour
```

4. Segments et formats RDP

Les segments envoyés par la couche application sont encapsulés dans les en-têtes par les couches transport, internet et réseau, comme suit :

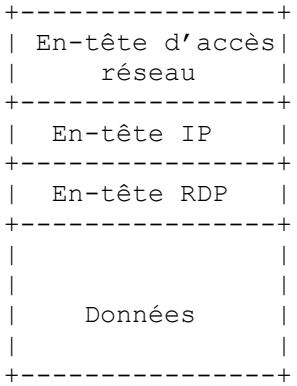

Figure 4 : Format de segment

4.1 Format d'en-tête IP

Lorsque ils sont utilisés dans l'environnement Internet, les segments RDP sont envoyés en utilisant l'en-tête IP version 4 comme décrit dans la RFC791, "Protocole Internet". Le numéro du protocole RDP est un (en décimal). Le champ Durée de vie devrait être réglé à une valeur raisonnable pour le réseau.

Tous les autres champs devraient être réglés comme spécifié dans la RFC-791.

4.2 Format d'en-tête RDP

Chaque segment RDP est précédé d'un en-tête RDP. Le format de l'en-tête est montré à la Figure 5 ci-dessous. L'en-tête RDP est de longueur variable et sa taille est indiqué par un champ d'une localisation fixée dans l'en-tête.

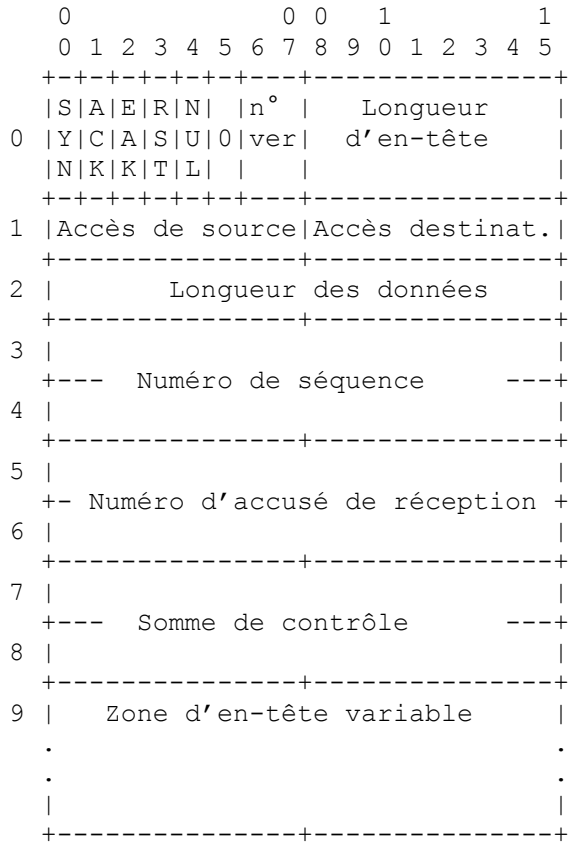

Figure 5 : Format d'en-tête RDP

4.2.1 Champs d'en-tête RDP

Fanions de contrôle
Ce champ de 8 bits occupe le premier octet du mot un de l'en-tête. Il est codé en binaire avec les bits suivants actuellement définis :

N° de bit	Nom du bit	Description
0	SYN	Établit la connexion et synchronise les numéros de séquence.
1	ACK	Le champ Accusé de réception est significatif.
2	EACK	Accusés de réception non cumulatifs (Étendus).
3	RST	Rétablissement de la connexion.
4	NUL	C'est un segment nul (données de longueur zéro).
5	Non utilisé	

Noter que les fanions SYN et RST sont envoyés comme segments séparés et ne peuvent pas contenir de données. Le fanion ACK peut accompagner tout message. Le segment NUL doit avoir une longueur de données de zéro, mais peut être accompagné par des informations de ACK et de EACK. Les autres bits de contrôle ne sont pas utilisés actuellement et sont définis comme étant à zéro.

Numéro de version
Ce champ occupe les bits 6-7 du premier octet. Il contient le numéro de version du protocole décrit par le présent document. La valeur actuelle est un (1).

Longueur d'en-tête
C'est la longueur de l'en-tête RDP en unités de deux (2) octets, incluant ce champ. Ce champ permet à RDP de trouver le début du champ Données, selon un pointeur sur la tête du segment. Ce champ fait 8 bits. Pour un segment sans section d'en-tête variable, le champ Longueur d'en-tête aura la valeur 9.

Accès de source et de destination
Les accès de source et de destination sont utilisés pour identifier les processus dans les deux hôtes qui sont en communication l'un avec l'autre. La combinaison des identifiants d'accès avec les adresses de source et de destination dans l'en-tête de protocole d'accès réseau sert à qualifier pleinement la connexion et constitue l'identifiant de connexion. Cela permet à RDP de distinguer plusieurs connexions entre deux hôtes. Chaque champ fait 8 bits, ce qui permet des numéros d'accès de 0 à 255 (en décimal).

Longueur des données
C'est la longueur en octets des données dans ce segment. La longueur des données n'inclut pas l'en-tête RDP. Ce champ fait 16 bits.
Numéro de séquence
C'est le numéro de séquence de ce segment. Ce champ fait 32 bits.

Numéro d'accusé de réception
Si le bit ACK est établi dans l'en-tête, c'est le numéro de séquence du segment que l'envoyeur de ce segment a reçu correctement en séquence en dernier. Une fois qu'une connexion est établie cela devrait toujours être envoyé. Ce champ fait 32 bits.

Somme de contrôle
Ce champ est une somme de contrôle de 32 bits de l'en-tête et des données du segment. La description de l'algorithme ci-dessous inclut deux variables, l'accumulateur de somme de contrôle et le pointeur de somme de contrôle. L'accumulateur de somme de contrôle est un registre réel de 32 bits dans lequel est formée la somme de contrôle. Le pointeur de somme de contrôle est inclus pour les besoins de la description, pour représenter l'opération d'avancée à travers les données par quatre octets (32 bits) à la fois. Une mise en œuvre n'a pas besoin de le conserver dans un registre.

1) Le pointeur de somme de contrôle est réglé à zéro, pour correspondre au début de la zone objet de la somme de contrôle. L'accumulateur de somme de contrôle est aussi initialisé à zéro avant de commencer le calcul de la somme de contrôle.

2) Le mot de mémoire de 32 bits situé à l'adresse référencée par le pointeur de somme de contrôle est ajouté arithmétiquement à l'accumulateur de somme de contrôle. Tout report propagé en dehors de l'accumulateur de somme de contrôle est ignoré. Le champ Somme de contrôle est lui-même remplacé par des zéros lorsque il est ajouté à l'accumulateur de somme de contrôle.

3) L'accumulateur de somme de contrôle subit une rotation à gauche d'une position binaire. Le pointeur de somme de contrôle est avancé pour correspondre à l'adresse du prochain mot de 32 bis dans le segment.

4) Les étapes 2 et 3 sont répétées jusqu'à ce que le segment entier ait été additionné. Si un segment contient un nombre d'octets d'en-tête et de données qui n'est pas un multiple entier de 4 octets, le dernier octet est bourré à droite avec des zéros pour former une quantité de 32 bits pour les besoins du calcul.

Zone d'en-tête variable
Cette zone est utilisée pour transmettre des paramètres pour les segments SYN et EACK.

4.3 Segment SYN

Le segment SYN est utilisé pour établir une connexion et synchroniser les numéros de séquence entre les deux hôtes. Le segment SYN contient aussi des informations pour l'hôte distant sur le nombre maximum de segments que le RDP local veut accepter et la taille maximum de segment qu'il peut accepter. Le segment SYN peut être combiné avec un ACK dans un segment mais n'est jamais combiné avec des données d'usager.

4.3.1 Format de segment SYN

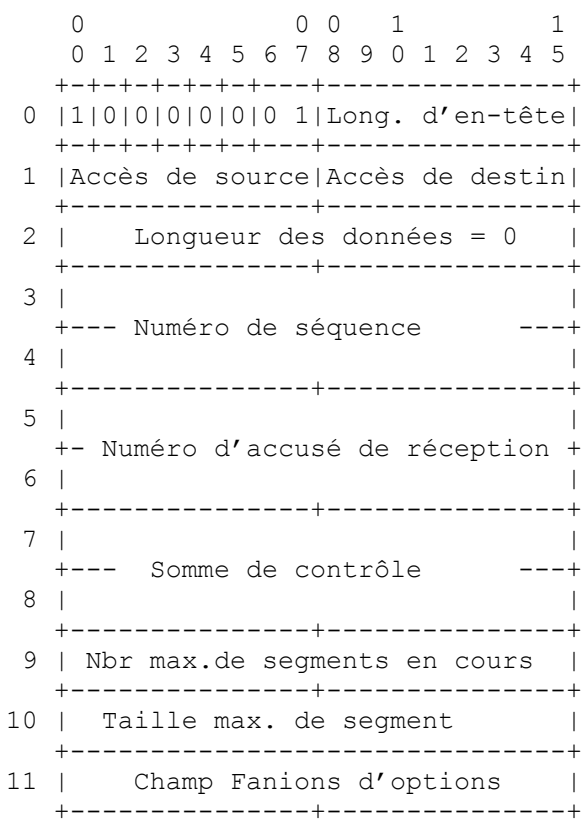

Figure 6 : Format de segment SYN

4.3.2 Champs de segment SYN

Numéro de séquence
Contient le numéro de séquence initial choisi pour cette connexion.

Numéro d'accusé de réception
Ce champ n'est valide que si le fanion ACK est établi. Dans ce cas, ce champ va contenir le numéro de séquence du segment SYN reçu de l'autre RDP.

Nombre maximum de segments en instance
C'est le nombre maximum de segments qui devraient être envoyés sans obtenir d'accusé de réception. C'est utilisé par le receveur comme moyen de contrôle de flux. Le nombre est choisi durant l'initialisation de la connexion et ne peut pas être changé ensuite pendant la durée de vie de la connexion.

Taille maximum de segment
Taille maximum de segment en octets que l'envoyeur devrait envoyer. Elle informe l'envoyeur de la taille des mémoires tampon du receveur. La taille spécifiée inclut la longueur de l'en-tête IP, de l'en-tête RDP, et des données. Elle n'inclut pas la longueur de l'en-tête de la couche d'accès réseau.

Champ Fanions d'options
Ce champ de deux octets contient un ensemble de fanions d'options qui spécifient l'ensemble des fonctions facultatives qui sont désirées pour cette connexion. Les fanions sont définis comme suit :

N° de bit	Nom du bit	Description
0	SDM	Mode de livraison en séquence.

Le Fanion Mode de livraison en séquence spécifie si la livraison des segments à l'utilisateur est en séquence (livré dans l'ordre des numéros de séquence) ou non en séquence (livré dans l'ordre d'arrivée, sans considération du numéro de séquence). Une valeur 0 spécifie la livraison des segments non en séquence, et une valeur de 1 spécifie la livraison en séquence.

4.4 Segment ACK

Le segment ACK est utilisé pour accuser réception de segments en séquence. Il contient à la fois le prochain numéro de séquence envoyé et le numéro de séquence d'accusé de réception dans l'en-tête RDP. Le segment ACK peut être envoyé comme segment séparé, mais il devrait être combiné avec des données chaque fois que possible. Les segments de données doivent toujours inclure le bit ACK et le champ Numéro d'accusé de réception.

4.4.1 Format de segment ACK

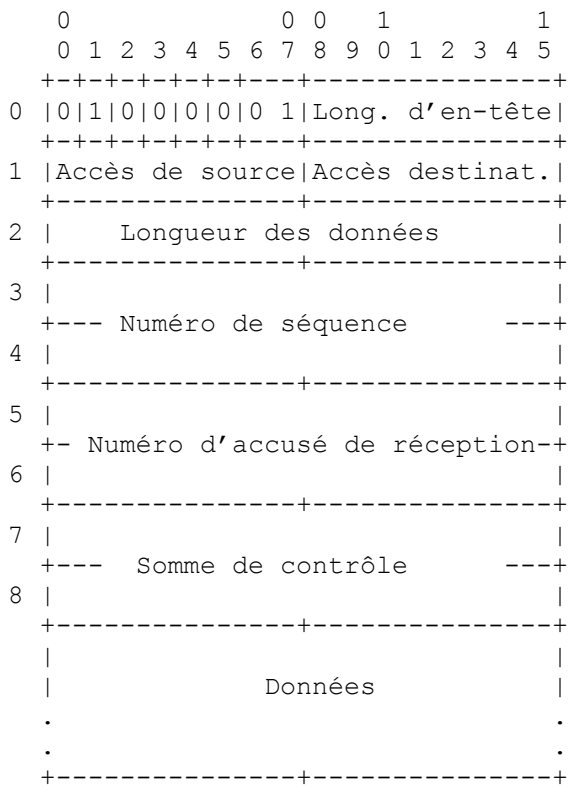

Figure 7 : Format de segment ACK

4.4.2 Champs de segment ACK

Longueur des données
Un champ Longueur des données différent de zéro indique que des données sont présentes dans le segment.

Numéro de séquence
La valeur du champ Numéro de séquence n'est avancée au prochain numéro de séquence que si des données sont présentes dans le segment. Un segment ACK sans données n'utilise pas d'espace de numéro de séquence.

Numéro d'accusé de réception
Le champ Numéro d'accusé de réception contient le numéro de séquence du dernier segment reçu dans l'ordre séquentiel.

4.5 Segment ACK étendu

Le segment EACK est utilisé pour acquitter les segments reçus hors séquence. Il contient les numéros de séquence d'un ou plusieurs segments reçus avec une somme de contrôle correcte, mais hors séquence. Le EACK est toujours combiné avec un ACK dans le segment, donnant le numéro de séquence du dernier segment reçu en séquence. Le segment EACK peut aussi inclure des données d'utilisateur.

4.5.1 Format de segment EACK

Le segment EACK a le format indiqué à la Figure 8.

4.5.2 Champs de segment EACK

Longueur des données
Un champ Longueur des données différent de zéro indique que des données sont présentes dans le segment.

Numéro de séquence
La valeur du champ Numéro de séquence n'est avancée au prochain numéro de séquence que si des données sont présentes dans le segment. Un segment EACK sans données n'utilise pas l'espace de numéros de séquence.

Numéro d'accusé de réception
Le champ Numéro d'accusé de réception contient le numéro de séquence du dernier segment reçu dans l'ordre séquentiel.

Numéro de séquence bien reçu
Chaque entrée est le numéro de séquence d'un segment qui a été reçu avec une somme de contrôle correcte, mais hors séquence.

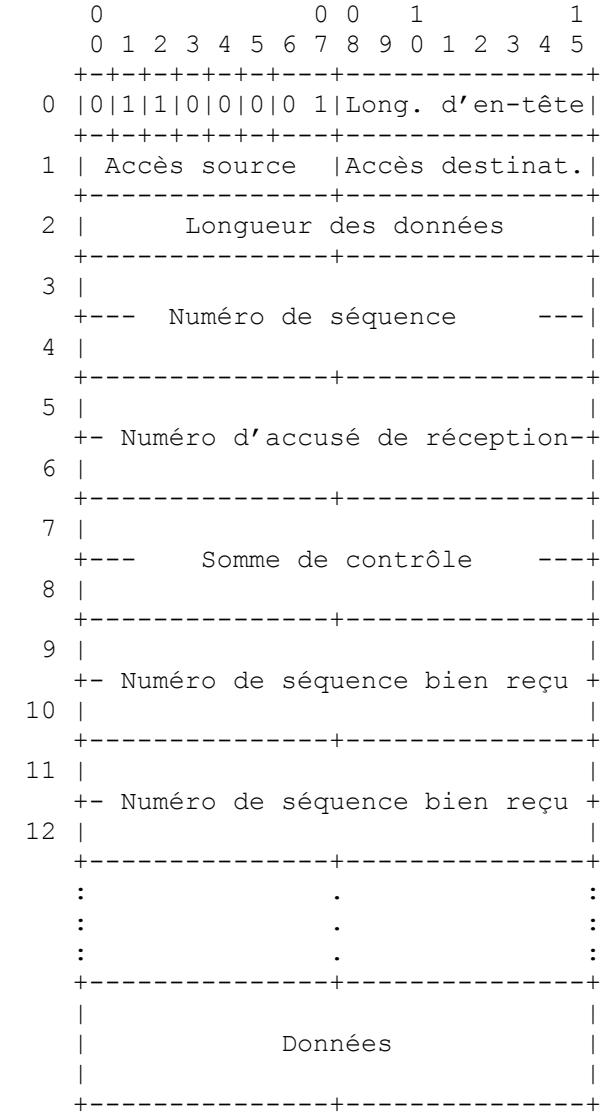

Figure 8 : Format de segment EACK

4.6 Segment RST

Le segment RST est utilisé pour fermer ou rétablir une connexion. À réception d'un segment RST, l'envoyeur doit arrêter d'envoyer et doit interrompre toutes les demandes non servies. Le segment RST est envoyé comme segment séparé et ne comporte aucune donnée.

4.6.1 Format du segment RST

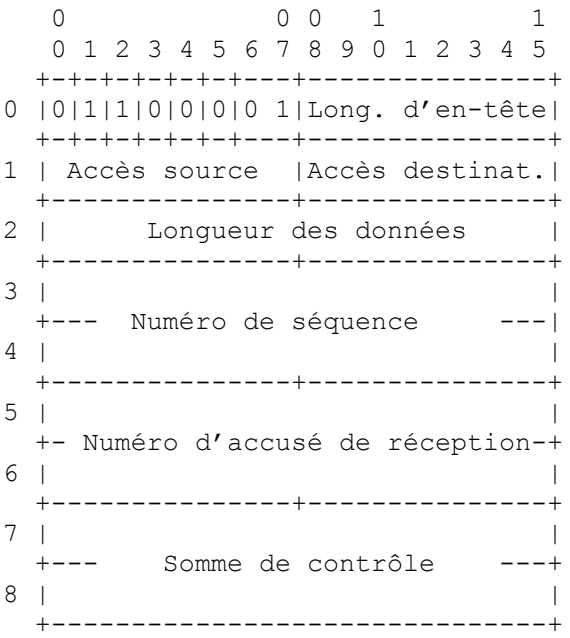

Figure 9 : Format de segment RST

4.7 Segment NUL

Le segment NUL est utilisé pour déterminer si l'autre côté d'une connexion est toujours actif. Lorsque un segment NUL est reçu, une mise en œuvre de RDP doit accuser réception du segment si une connexion valide existe et si le numéro de séquence du segment tombe dans la fenêtre d'acceptation. Le segment est alors éliminé. Le segment NUL peut être combiné avec un ACK dans un segment mais n'est jamais combiné avec des données d'utilisateur.

4.7.1 Format de segment NUL

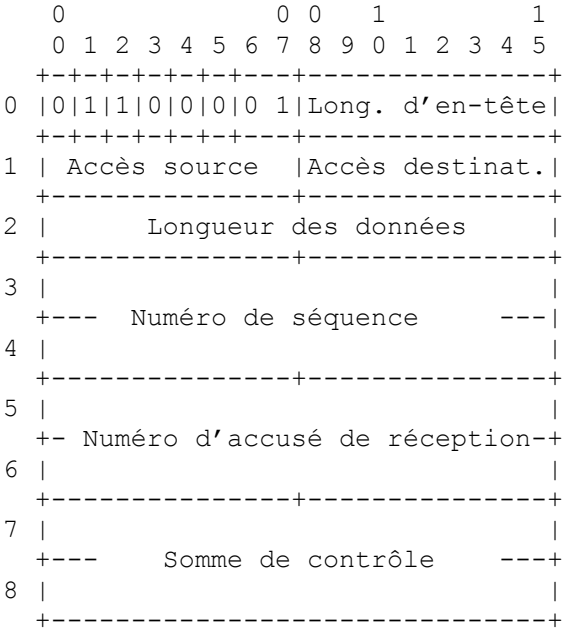

Figure 10 : Format de segment NUL

5 Exemples de fonctionnement

5.1 Établissement de connexion

Voici l'exemple d'une connexion qu'on établit entre l'hôte A et l'hôte B. L'hôte B a fait un Open passif et il est dans l'état LISTEN. L'hôte A fait un Open actif pour établir la connexion.

Temps	État	Hôte A	Hôte B	État
1.	CLOSED			LISTEN
2	SYN-SENT	<SEQ=100><SYN> --->		
3.			<--- <SEQ=200><ACK=100><SYN,ACK>	SYN-RCVD
4.	OPEN	<SEQ=101><ACK=200> --->		OPEN
5.		<SEQ=101><ACK=200><Données> --->		
6.			<--- <SEQ=201><ACK=101>	

5.2 Établissement de connexions simultanées

Voici l'exemple de deux hôtes qui essayent d'établir des connexions l'un avec l'autre au même moment. L'hôte A envoie une demande SYN à l'hôte B au moment où l'hôte B envoie une demande SYN à l'hôte A.

Temps	État	Hôte A	Hôte B	État
1.	CLOSED			CLOSED
2.	SYN-SENT	<SEQ=100><SYN> --->		
			<--- <SEQ=200><SYN>	SYN-SENT
3.	SYN-RCVD			SYN-RCVD
		<SEQ=100><ACK=200><SYN,ACK> --->		
			<--- <SEQ=200><ACK=100><SYN,ACK>	
4.	OPEN			OPEN

5.3 Segments perdus

Voici un exemple de ce qui arrive lorsque un segment est perdu. Il montre comment les segments peuvent être acquittés hors de séquence et que seul le segment manquant a besoin d'être retransmis. Noter que dans cet exemple et les suivants "EA" signifie "Accusé de réception hors séquence."

Temps	Hôte A	Hôte B
1.	<SEQ=100><ACK=200><Données> --->	
2.		<--- <SEQ=201><ACK=100>
3.	<SEQ=101><ACK=200><Données> (segment perdu)	
4.		
5.	<SEQ=102><ACK=200><Données> --->	
6.		<-- <SEQ=201><ACK=100><EA=102>
7.	<SEQ=103><ACK=200><Données> --->	
8.		<--- <SEQ=201><ACK=100><EA=102,103>
9.	<SEQ=101><ACK=200><Données> --->	
10.		<--- <SEQ=201><ACK=103>
11.	<SEQ=104><ACK=200><Données> --->	
12.		<--- <SEQ=201><ACK=104>

5.4 Segments reçus dans le désordre

Voici un exemple de segments reçus dans le désordre. Il illustre mieux l'utilisation des accusés de réception de segments déclassés pour empêcher des retransmissions inutiles.

Temps	Hôte A	Hôte B
1.	<SEQ=100><ACK=200><Données> --->	
2.		<--- <SEQ=201><ACK=100>
3.	<SEQ=101><ACK=200><Données> (retardé)	
4.		
5.	<SEQ=102><ACK=200><Données> --->	
6.		<--- <SEQ=201><ACK=100><EA=102>
7.	<SEQ=103><ACK=200><Données> --->	
	---> (le segment retardé 101 arrive)	
8.		<--- <SEQ=201><ACK=103>
9.	<SEQ=104><ACK=200><Données> --->	
10.		<--- <SEQ=201><ACK=104>

5.5 Communication sur chemin à long délai

Voici un exemple de transfert de données sur un chemin à fort retard. Dans cet exemple, il est permis à l'hôte A d'avoir jusqu'à cinq segments non acquittés. L'exemple montre qu'il n'est pas nécessaire d'attendre un accusé de réception pour envoyer des données supplémentaires.

Temps	Hôte A	Hôte B
1.	<SEQ=100><ACK=200><Données> -1 --->	
2.	<SEQ=101><ACK=200><Données> -2->	
3.	<SEQ=102><ACK=200><Données> -3-> -1-> (reçu)	
4.		<---4- <SEQ=201><ACK=100>
5.	<SEQ=103><ACK=200><Données> -5-> -2-> (reçu)	
6.		<-6- <SEQ=201><ACK=101>
7.	<SEQ=104><ACK=200><Données> -7-> -3-> (reçu)	
8.		<-8- <SEQ=201><ACK=102> (reçu) <-4-
9.	<SEQ=105><ACK=200><Données> -9-> -5-> (reçu)	
10.		<-10- <SEQ=201><ACK=103> (reçu) <-6-
11.	<SEQ=106><ACK=200><Données> -11-> -7-> (reçu)	
12.		<-12- <SEQ=201><ACK=104> (reçu) <-8-
13.	-9-> (reçu)	
14.		<-13- <SEQ=201><ACK=105> (reçu) <-10-
15.	-11-> (reçu)	
16.		<-14- <SEQ=201><ACK=106> (reçu) <-12-
17.		(reçu) <-13-
18.		(reçu) <-14-

5.6 Communication sur chemin à long délai avec pertes de segments

Voici un exemple de communication sur un chemin à long délai avec un segment perdu. Il montre qu'en accusant réception des segments hors séquence, seul le segment perdu a besoin d'être retransmis.

Temps	Hôte A	Hôte B
1.	<SEQ=100><ACK=200><Données> -1 --->	
2.	<SEQ=101><ACK=200><Données> -2->	
3.	<SEQ=102><ACK=200><Données> -3-> -1-> (reçu)	
4.		<---4- <SEQ=201><ACK=100>
5.	<SEQ=103><ACK=200><Données> (segment perdu) -2-> (reçu)	
6.		<-5- <SEQ=201><ACK=101>
7.	<SEQ=104><ACK=200><Données> -6-> -3-> (reçu)	
8.		<-7- <SEQ=201><ACK=102> (reçu) <-4-
9.	<SEQ=105><ACK=200><Données> -8->	
10.		(reçu) <-5-
11.	<SEQ=106><ACK=200><Données> -10-> -6-> (reçu)	
12.	-8-> (reçu)	<-11- <SEQ=201><ACK=102><EA=104> (reçu) <-7-
13.	-10-> (reçu)	<-12- <SEQ=201><ACK=102><EA=104,105>
14.		<-13- <SEQ=201><ACK=102><EA=104-106> (reçu) <-11-
15.	<SEQ=103><ACK=200><Données> -14->	(reçu) <-12-
16.	-14-> (reçu)	(reçu) <-13-
17.		<-15- <SEQ=201><ACK=106>
18.		
19.		(reçu) <-15-

5.7 Détection d'une connexion semi ouverte sur récupération de panne

Voici un exemple d'un hôte qui détecte une connexion semi ouverte due à la panne et au redémarrage subséquent de l'hôte. Dans cet exemple, l'hôte A tombe en panne durant une session de communication, puis récupère et essaye de rouvrir la connexion. Durant la tentative de réouverture, il découvre qu'il existe toujours une connexion semi ouverte et il rétablit alors l'autre côté. Les deux côtés étaient dans l'état OPEN avant la panne.

Temps	État	Hôte A	Hôte B	État
1.	OPEN (panne !)		<--- <SEQ=200><ACK=100><ACK>	OPEN
2.	CLOSED (récupère)			OPEN

3.	SYN-SENT	<SEQ=400><SYN> --->	(?)	OPEN	
4.	SYN-SENT	(!)	<--- <SEQ=200><ACK=100><ACK>	OPEN5.	SYN-SENT
	<SEQ=101><RST> --->	OPEN (interrompre)			
6.	SYN-SENT			CLOSED	
7.	SYN-SENT	<SEQ=400><SYN> --->			

5.8 Détection d'une connexion semi ouverte de la part du côté actif

Voici un autre exemple de détection d'une connexion semi ouverte à cause de la panne et du redémarrage d'un hôte impliqué dans une connexion. Dans cet exemple, l'hôte A tombe encore en panne et redémarre. L'hôte B est encore actif et essaye d'envoyer des données à l'hôte A. Comme l'hôte A ignore tout de la connexion, il rejette les données avec un segment RST, causant le rétablissement de la connexion par l'hôte B.

Temps	État	Hôte A	Hôte B	État
1.	(panne !)			OPEN
2.	CLOSED		<--- <SEQ=200><ACK=100><Données>	OPEN
3.	CLOSED	<SEQ=101><RST> --->		(interrompre)
4.	CLOSED			CLOSED

Appendice A Mise en œuvre d'un RDP minimal

Il n'est pas nécessaire de mettre en œuvre la totalité de la spécification de RDP pour être capable d'utiliser RDP. Un sous-ensemble de RDP peut être utilisé par des applications simples telles qu'un chargeur, où la taille du module de protocole peut être importante. Par exemple, une mise en œuvre de RDP pour le téléchargement peut employer les restrictions suivantes :

- Une seule connexion et un seul enregistrement de connexion sont pris en charge. C'est la connexion utilisée pour charger l'appareil.
- Un seul accès bien connu est utilisé comme accès de chargement. Les accès allouables ne sont pas mis en œuvre.
- Seule la demande Open passive est mise en œuvre. Les demandes Open actives ne sont pas mises en œuvre.
- L'option de livraison en séquence n'est pas prise en charge. Les messages qui arrivent décalés sont livrés dans l'ordre dans lequel ils arrivent.
- Si l'efficacité est moins importante que la taille du protocole, le dispositif d'accusé de réception étendu n'a pas besoin d'être pris en charge.

Index

SND.UNA
somme de contrôle *(checksum)*
STATE
SYN
traitement d'événement
taille maximum de segment
TCP