Groupe de travail Réseau
Request for Comments : 1122
STD 3
Traduction Claude Brière de L'Isle

Internet Engineering Task Force
R. Braden, éditeur
octobre 1989

Exigences pour les hôtes Internet – Couches de communication

Statut du présent mémoire

La présente RFC est une spécification officielle pour la communauté de l'Internet. Elle incorpore par références, amende, corrige et complète les documents de normalisation du protocole principal qui se rapportent aux hôtes. La distribution du présent document n'est soumise à aucune restriction.

Résumé

La présente RFC fait partie d'une paire de documents qui définissent et discutent les exigences pour les logiciels des hôtes Internet. Elle traite des couches de protocole de communication : couche de liaison, couche IP , et couche transport ; sa compagne est la RFC 1123 qui traite des protocoles d'application et de prise en charge.

Table des matières

1. Introduction

Le présent document fait partie d'une paire de documents qui définissent et exposent les exigences pour les mises en œuvre de système d'hôte de la suite des protocoles de l'Internet. La présente RFC couvre les couches de protocole de communication : couche de liaison, couche IP, et couche de transport. Sa RFC associée, "Exigences pour les hôtes Internet -- Application et prise en charge" [RFC1123], couvre les protocoles de la couche d'application. Le présent document devrait aussi être lu en conjonction avec "Exigences pour les passerelles Internet" [RFC1009].

Ces documents sont destinés à fournir des lignes directrices aux fabricants, développeurs et utilisateurs des logiciels de communication Internet. Ils représentent le consensus d'un large corpus d'expérience technique et de réflexion, auquel ont contribué les membres des communautés Internet de la recherche et de l'industrie.

La présente RFC énumère les protocoles standard que doit utiliser un hôte connecté à l'Internet, et elle incorpore par référence les RFC et autres documents qui décrivent les spécifications actuelles de ces protocoles. Elle corrige des erreurs dans les documents référencés et ajoute des exposés et lignes directrices supplémentaires pour un développeur.

Pour chaque protocole, le présent document contient aussi un ensemble explicite d'exigences, recommandations, et options. Le lecteur doit comprendre que la liste des exigences de ce document est incomplète par elle-même ; l'ensemble complet des exigences pour un hôte Internet est principalement défini dans les documents de spécification de protocole standard, avec les corrections, amendements, et suppléments contenus dans la présente RFC.

Une mise en œuvre de bonne foi des protocoles qui ont été produits après une lecture attentive de la RFC et avec une certaine interaction avec la communauté technique de l'Internet, et qui suit de bonnes pratiques d'ingénierie de logiciel de communications, ne devrait différer que de façon mineure des exigences du présent document. Et donc, dans de nombreux cas, les "exigences" de la présente RFC sont déjà établies ou impliquées dans les documents de protocole standard, de sorte que leur inclusion ici est, en un sens, redondante. Cependant, elles ont été incluses parce que certaines mises en œuvre ont fait dans le passé de mauvais choix, causant des problèmes d'interopérabilité, de performance, et/ou de robustesse.

Le présent document inclut des discussions et explications sur beaucoup des exigences et recommandations. Une simple liste des exigences serait dangereuse, parce que:

- Certaines dispositions exigées sont plus importantes que d'autres, et certaines dispositions sont facultatives.

- Il peut y avoir des raisons valides pour que le produit d'un fabricant particulier conçu pour un contexte restrictif choisisse d'utiliser des spécifications différentes.

Cependant, les spécifications du présent document doivent être suivies pour satisfaire au but général d'inter opération d'hôtes choisis arbitrairement à travers la diversité et la complexité du système Internet. Bien que la plupart des mises en œuvre actuelles ne réussissent pas à satisfaire de diverses façons à ces exigences, certaines mineures, certaines majeures, la présente spécification est l'idéal vers lequel on doit tendre.

Ces exigences se fondent sur le niveau actuel de l'architecture Internet. Le présent document sera mis à jour en tant que de besoin pour fournir des éclaircissements supplémentaires ou pour inclure des informations additionnelles dans les domaines dans lesquels les spécifications évoluent encore.

Cette section d'introduction commence par un bref aperçu de l'architecture de l'Internet dans ses rapports avec les hôtes, puis donne quelques conseils généraux aux fabricants de logiciels d'hôtes. Finalement, y figurent quelques lignes directrices sur la façon de lire le reste du document et un peu de terminologie.

1.1 Architecture de l'Internet

Les fondements généraux et l'exposé de l'architecture de l'Internet et la prise en charge de la suite des protocoles se trouvent dans le Manuel des protocoles du DDN DDN-NIC] ; pour les fondements, voir par exemple [INTRO:9], [INTRO:10], et [INTRO:11]. La référence [RFC0980] décrit les procédure pour l'obtention des documents de protocole de l'Internet, alors que [RFC1700] contient la liste des numéros alloués au sein des protocoles de l'Internet.

1.1.1 Hôtes Internet

Un ordinateur hôte, ou simplement un "hôte," est le consommateur ultime des services de communication. Un hôte exécute généralement des programmes d'application au nom d'un utilisateur (ou plusieurs), employant des services de communication réseau et/ou Internet à l'appui de cette fonction. Un hôte Internet correspond au concept de "Système d'extrémité" utilisé dans la suite de protocoles OSI [RFC0995].

Un système de communication Internet consiste en réseaux de paquets interconnectés qui prennent en charge la communication parmi les ordinateurs hôtes utilisant les protocoles de l'Internet. Les réseaux sont interconnectés en utilisant des ordinateurs de commutation de paquets appelés "passerelles" ou "routeurs IP" par la communauté de l'Internet, et "Systèmes intermédiaires" dans le monde OSI [RFC0995]. La RFC "Exigences pour les passerelles Internet" [RFC1009] contient les spécifications officielles des passerelles Internet. Cette RFC, conjointement avec le présent document et son compagnon [RFC1123] définit les règles de la réalisation actuelle de l'architecture Internet.

Les hôtes Internet couvrent une large gamme de tailles, vitesses, et fonctions. Leurs tailles vont du petit microprocesseur au supercalculateur en passant par la station de travail et l'ordinateur . Leurs fonctions vont des hôtes dédiés à une seule fin (comme les serveurs terminaux) aux hôtes de plein exercice qui prennent en charge une diversité de services réseau en ligne, qui comportent normalement la connexion à distance, le transfert de fichiers, et la messagerie électronique.

Un hôte est généralement dit à rattachements multiples si il a plus d'une interface sur le même réseau ou sur différents réseaux. Voir le paragraphe 1.1.3 "Terminologie".

1.1.2 Hypothèses architecturales

L'architecture actuelle de l'Internet se fonde sur un ensemble d'hypothèses sur le système de communications. Les hypothèses les plus pertinentes pour les hôtes sont les suivantes :

(a) L'Internet est un réseau de réseaux.
Chaque hôte est directement connecté à un ou des réseaux particuliers ; sa connexion à l'Internet est seulement conceptuelle. Deux hôtes sur le même réseau communiquent l'un avec l'autre en utilisant le même ensemble de protocoles qu'ils utiliseraient pour communiquer avec des hôtes sur des réseaux distants.

(b) Les passerelles ne conservent pas les informations d'état de connexion.
Pour améliorer la robustesse du système de communication, les passerelles sont conçues pour être sans état, transmettant chaque datagramme IP indépendamment des autres datagrammes. Il en résulte que les chemins redondants peuvent être exploités pour fournir un service robuste en dépit des défaillances des passerelles et réseaux intervenants.

Toutes les informations d'état nécessaires pour le contrôle et la fiabilité de bout en bout sont mises en œuvre dans les hôtes, dans la couche de transport ou dans les programmes d'application. Toutes les informations de contrôle de connexion sont donc colocalisées avec les points d'extrémité de la communication, aussi ne seront elles perdues que sur défaillance du point d'extrémité.

(c) La complexité de l'acheminement devrait être dans les passerelles.
L'acheminement est un problème complexe et difficile, qui devrait être effectué par les passerelles, et non par les hôtes. Un objectif important est d'isoler le logiciel d'hôte des changements causés par l'inévitable évolution de l'architecture d'acheminement de l'Internet.

(d) Le système doit tolérer de larges variations du réseau.
Un objectif de base de la conception de l'Internet est de tolérer une large gamme de caractéristiques de réseau – par exemple, de bande passante, de délai, de perte de paquet, de réordonnancement de paquet, et de taille maximale de paquet. Un autre objectif est la robustesse aux défaillances des réseaux, passerelles et hôtes individuels, en utilisant toute la bande passante encore disponible. Finalement, le but est une pleine "interconnexion de système ouvert" : un hôte Internet doit être capable d'interopérer de façon robuste et effective avec tout autre hôte de l'Internet, à travers les divers chemins de l'Internet.

Parfois, les développeurs d'hôtes ont conçu des objectifs moins ambitieux. Par exemple, l'environnement de LAN est normalement beaucoup plus bénin que l'Internet global ; les LAN ont peu de perte de paquet et de retard et ne réorganisent pas les paquets. Certains fabricants ont mis sur le marché des mises en œuvre d'hôtes qui sont adéquates pour un simple environnement de LAN, mais fonctionnent assez mal dans une interopération générale. Le fabricant justifie un tel produit par son faible coût au sein du marché restreint du LAN. Cependant, les LAN isolés restent rarement isolés longtemps ; ils sont rapidement relayés les uns les autres par des passerelles jusqu'à des internets à la dimension de l'organisation, et finalement jusqu'au système Internet mondial. Finalement, ni le consommateur ni le fabricant ne sont vraiment satisfaits par des logiciels d'hôte Internet incomplets ou inférieurs aux normes.

Les exigences affichées dans le présent document sont conçues pour un hôte Internet assumant la totalité de ses fonctions, capable d'interopération complète sur tout chemin Internet choisi arbitrairement.

1.1.3 Suite des protocoles de l'Internet

Pour communiquer en utilisant le système Internet, un hôte doit mettre en œuvre l'ensemble de protocoles ordonné en couches qui comprend la suite des protocoles de l'Internet. Un hôte doit normalement mettre en œuvre au moins un protocole de chaque couche.

Les couches de protocole utilisées dans l'architecture de l'Internet sont les suivantes [RFC1011] :

- Couche d'application
  La couche d'application est la couche supérieure de la suite des protocoles de l'Internet. La suite Internet ne fait pas d'autre subdivision de la couche d'application, bien que certains des protocoles de couche d'application de l'Internet contiennent une répartition interne en sous-couches. La couche d'application de la suite Internet combine essentiellement les fonctions des deux couches supérieures -- Présentation et Application – du modèle de référence OSI.

  On distingue deux catégories de protocoles de couche d'application : les protocoles d'utilisateur qui fournissent le service directement aux usagers, et les protocoles de soutien qui fournissent des fonctions système communes. Les exigences pour les protocoles d'utilisateur et celles pour les protocoles de soutien se trouvent dans la [RFC1123].

  Les protocoles d'utilisateur Internet les plus communs sont :
  - Telnet (connexion distante)
  - FTP (transfert de fichier)
  - SMTP (livraison de messagerie électronique)

  Il y a un grand nombre d'autres protocoles d'utilisateur normalisés [RFC1011] et de protocoles d'utilisateur privés.

  Les protocoles de soutien, utilisés pour la transposition des noms d'hôte, l'amorçage, et la gestion, incluent les protocoles SNMP, BOOTP, RARP, et du système des noms de domaine (DNS).

- Couche transport
  La couche transport fournit un service de communication de bout en bout pour les applications. À présent, il y a deux protocoles principaux de couche transport :
  - Protocole de commande de transmission (TCP)
  - Protocole de datagrammes d'utilisateur (UDP)

TCP est un service de transport fiable orienté connexion qui fournit la fiabilité de bout en bout, le re-séquençage, et le contrôle des flux. UDP est un service de transport sans connexion (ou par "datagramme").

D'autres protocoles de transport ont été développés par la communauté de la recherche, et l'ensemble des protocoles officiels de transport de l'Internet pourra être étendu à l'avenir.

Les protocoles de la couche transport sont exposés à la Section 4.

- Couche Internet

  Tous les protocoles de transport Internet utilisent le protocole Internet (IP) pour porter les données de l'hôte de source à l'hôte de destination. IP est un service inter-réseau sans connexion ou par datagramme, qui ne fournit pas de garantie de livraison de bout en bout. Et donc, les datagrammes IP peuvent arriver à l'hôte de destination endommagés, dupliqués, décalés, ou pas du tout. Les couches au-dessus d'IP sont chargées du service de livraison fiable lorsque c'est nécessaire. Le protocole IP comporte des dispositions pour l'adressage, la spécification du type de service, la fragmentation et le réassemblage, et les informations de sécurité.

  La nature par datagramme ou sans connexion du protocole IP est une caractéristique fondamentale de l'architecture de l'Internet. Le protocole Internet a été le modèle de réseau sans connexion pour le protocole OSI [RFC0994].

  ICMP est un protocole de commande qui est considéré comme une partie intégrante de IP, bien qu'il soit architecturalement sur une couche supérieure à IP, c'est-à-dire qu'il utilise IP pour porter ses données de bout en bout juste comme le fait un protocole de transport tel que TCP ou UDP. ICMP fournit des rapports d'erreurs, des rapports d'encombrement, et la redirection de passerelle de premier bond.

  IGMP est un protocole de couche Internet utilisé pour établir des groupes d'hôtes de façon dynamique pour la diffusion groupée IP.

  Les protocoles de couche Internet IP, ICMP, et IGMP sont exposés à la Section 3.

- Couche de liaison

  Pour communiquer sur le réseau sur lequel il est directement connecté, un hôte doit mettre en œuvre le protocole de communication utilisé pour s'interfacer à ce réseau. On appelle cela un protocole de couche de liaison ou de couche d'accès au support.

  Il y a une grande variété de protocoles de couche de liaison, correspondant aux nombreux différents types de réseaux. Voir la Section 2.

1.1.4 Code de passerelle incorporée

Certains hôtes Internet incluent une fonctionnalité de passerelle incorporée, de sorte que ces hôtes peuvent transmettre des paquets comme le ferait une passerelle, tout en effectuant les fonctions de couche d'application d'un hôte.

De tels systèmes bi-fonctionnels doivent suivre les exigences de la RFC sur les exigences des hôtes [RFC1009] en ce qui concerne leurs fonctions de passerelle, et doivent suivre le présent document en ce qui concerne leurs fonctions d'hôte. Dans tous les cas de chevauchement, les deux spécifications devraient être en cohérence.

Il y a des opinions diverses dans la communauté de l'Internet sur les fonctionnalités de passerelle incorporée. Les principaux arguments sont les suivants :

Pour : Dans un environnement de réseau local ou le réseautage est informel, ou sur des internets isolés, il peut être pratique et économique d'utiliser les systèmes d'hôte existants comme passerelle. Il y a aussi un argument architectural en faveur de la fonctionnalité de passerelle incorporée : le multi-rattachement est plus courant qu'il n'était prévu à l'origine, et le multi-rattachement force un hôte à prendre des décisions d'acheminement comme s'il était une passerelle. Si l'hôte à rattachements multiples contient une passerelle incorporée, il aura une pleine connaissance de l'acheminement, d'où il résulte qu'il sera capable de prendre des décisions d'acheminement optimales.

Contre : Les algorithmes et protocoles de passerelles sont encore en évolution et ils vont continuer d'évoluer avec la croissance du système Internet. Essayer d'inclure une fonction générale de passerelle au sein de la couche IP d'hôte va forcer les personnes chargées de la maintenance du système d'hôte à suivre de plus fréquents changements. Et aussi, un parc plus important de mises en œuvre de passerelles rendra la coordination des

changements plus difficile. Finalement, la complexité d'une passerelle de couche IP est un peu supérieure à celle d'un hôte, ce qui rend les tâches de mise en œuvre et de fonctionnement plus complexes.
De plus, le style de fonctionnement de certains hôtes n'est pas approprié à la fourniture d'un service de passerelle stable et robuste.

Ces points de vue ont tous deux des mérites considérables. On peut en tirer la conclusion qu'un administrateur d'hôte doit avoir une claire conscience de ce qu'un hôte agit ou non comme passerelle. Voir au paragraphe 3.1 les exigences détaillées.

1.2 Considérations générales

Deux importantes leçons tirées par les fabricants de logiciel d'hôtes Internet devraient être considérées avec attention par les nouveaux fabricants.

1.2.1 Continuer l'évolution de l'Internet

L'énorme croissance de l'Internet a révélé des problèmes de gestion et d'échelle dans les grands systèmes de communication par paquets fondée sur le datagramme. Ces problèmes sont en voie de résolution, et il en résulte une continuation de l'évolution des spécifications décrites dans le présent document. Ces changements seront soigneusement planifiés et contrôlés, car il y a une participation intensive des fabricants et des organisations chargées du fonctionnement des réseaux à cette planification.

Développement, évolution, et révision sont les caractéristiques des protocoles de réseau informatique d'aujourd'hui, et cette situation va persister pendant plusieurs années. Un fabricant qui développe des logiciels de communication informatique pour la suite des protocoles de l'Internet (ou toute autre suite de protocoles !) et échoue ensuite à entretenir et mettre à jour ce logiciel lorsque les spécifications changent, va laisser des cohortes de consommateurs mécontents. L'Internet est un grand réseau de communication, et les usagers sont en constant contact à travers lui. L'expérience montre que la connaissance des déficiences du logiciel d'un fabricant se propage rapidement à travers la communauté technique de l'Internet.

1.2.2 Principe de robustesse

À toutes les couches des protocoles, il y a une règle générale dont l'application peut conduire à d'énormes bénéfices en robustesse et en interopérabilité [RFC0791] :

"Soyez libéraux dans ce que vous acceptez, et conservateur dans ce que vous envoyez."

Un logiciel devrait être écrit de façon à traiter toutes les erreurs imaginables, même invraisemblables ; tôt ou tard, viendra un paquet avec cette combinaison particulière d'erreurs et d'attributs, et sauf si le logiciel y est préparé, le chaos peut s'ensuivre. En général, le mieux est de supposer que le réseau est plein d'entités malveillantes qui vont envoyer des paquets conçus pour avoir le pire effet possible. Cette hypothèse va conduire à des concepts de protection convenables, bien que les problèmes les plus sérieux sur l'Internet aient été causés par des mécanismes non envisagés déclenchés par des événements à faible probabilité ; la simple malice de l'homme n'aurait jamais pu suivre un cours aussi vicieux !

L'adaptabilité au changement doit être conçue à tous les niveaux du logiciel d'hôte Internet. Un simple exemple est de considérer une spécification de protocole qui contient une énumération de valeurs pour un champ d'en-tête particulier -- par exemple, un champ de type, un numéro d'accès, ou un code d'erreur ; cette énumération doit être supposée incomplète. Et donc, si une spécification de protocole définit quatre codes d'erreur possibles, le logiciel ne doit pas lâcher si un cinquième code apparaît. Un code indéfini pourrait être noté dans un journal de bord (voir ci-dessous), mais il ne doit pas causer de défaillance.

La seconde partie du principe est presque aussi importante : les logiciels sur les autres hôtes peuvent contenir des imperfections qui rendent peu raisonnable l'exploitation de dispositifs de protocole légaux mais obscurs. Il est malavisé de s'éloigner du simple et de l'évident, de peur que des effets malencontreux n'en résultent ailleurs. Un corollaire de ce conseil est de "surveiller les hôtes qui se conduisent mal" ; les logiciels d'hôtes devraient être prêts, non seulement à survivre à la présence d'hôtes au mauvais comportement, mais aussi à coopérer pour limiter la quantité de perturbations que de tels hôtes peuvent causer aux facilités de communications partagées.

1.2.3 Journalisation des erreurs

L'Internet inclut une grande variété de systèmes d'hôtes et de passerelles, chacun mettant en œuvre de nombreux protocoles et couches de protocole, dont certains contiennent des erreurs et des caractéristiques erronées dans le logiciel de protocole Internet. Par suite de la complexité, de la diversité, et de la distribution des fonctions, le diagnostic des problèmes de l'Internet est souvent très difficile.

Le diagnostic des problèmes sera facilité si la mise en œuvre d'hôte comporte une facilité bien conçue pour enregistrer les événements de protocole erronés ou "étranges". Il est important d'inclure autant d'informations de diagnostic que possible lors de l'enregistrement d'une erreur. En particulier, il est souvent utile d'enregistrer le ou les en-têtes d'un paquet qui a causé une erreur. Cependant, il faut veiller à s'assurer que l'enregistrement des erreurs ne consomme pas des quantités prohibitives de ressources ou n'interfère pas par ailleurs avec le fonctionnement de l'hôte.

Il y a une tendance à submerger les fichiers d'enregistrement d'erreur avec des événements de protocole anormaux mais anodins ; ceci peut être évité en utilisant un journal "circulaire", ou en ne permettant l'enregistrement que sur diagnostic d'une défaillance reconnue. Cela peut être utile de filtrer et compter les messages dupliqués successifs. Une stratégie qui semble bien fonctionner est de : (1) toujours compter les anomalies et de rendre un tel compte accessible par le protocole de gestion (voir [RFC1123]) ; et (2) de permettre sélectivement l'enregistrement d'une grande variété d'événements. Par exemple, il peut être utile d'avoir la capacité de "tout enregistrer" ou de "tout enregistrer pour l'hôte X".

Noter que des gestions différentes peuvent avoir des politiques qui diffèrent quant à la quantité d'enregistrements d'erreurs qu'elles veulent normalement permettre sur un hôte. Certains diront, "si cela ne me fait pas de tort, je ne veux pas en entendre parler", alors que d'autres voudront avoir une attitude plus attentive et agressive quant à la détection et la suppression des anomalies de protocole.

1.2.4 Configuration

L'idéal serait qu'une mise en œuvre d'hôte de la suite des protocoles Internet puisse être entièrement auto-configurable. Cela permettrait que la totalité de la suite soit mise en œuvre sur un CD-ROM ou gravée dans le silicium, cela simplifierait les stations de travail sans disque, et ce serait un immense avantage pour les administrateurs de LAN débordés aussi bien que pour les fabricants. Cet idéal n'a pas été atteint, et en fait nous en sommes loin.

En de nombreux points du présent document, on trouvera l'exigence qu'un paramètre soit une option configurable. Il y a plusieurs raisons à une telle exigence. Dans quelques cas, il y a actuellement une incertitude ou un désaccord sur ce qui serait la meilleure valeur, et il pourra être nécessaire de mettre à jour la valeur recommandée à l'avenir. Dans d'autres cas, la valeur dépend réellement de facteurs externes -- par exemple, de la taille de l'hôte et de la distribution de ses charges de communication, ou de la vitesse et de la topologie des réseaux voisins – et les algorithmes auto-réglables sont indisponibles et peut-être insuffisants. Dans certains cas, la possibilité de configurer est nécessaire à cause d'exigences administratives.

Finalement, certaines options de configuration sont nécessaires pour communiquer avec des mises en œuvre obsolètes ou incorrectes des protocoles, distribuées sans sources, qui persistent malheureusement dans de nombreuses parties de l'Internet. Pour faire coexister les systèmes corrects avec ces systèmes fautifs, les administrateurs ont souvent à "dis-configurer" les systèmes corrects. Ce problème se corrigera de lui-même graduellement avec l'éviction des systèmes fautifs, mais il ne peut être ignoré par les fabricants.

Lorsque nous disons qu'un paramètre doit être configurable, nous n'avons pas l'intention d'exiger que sa valeur soit lue explicitement à partir d'un fichier de configuration à chaque amorçage. Nous recommandons que les développeurs établissent une valeur par défaut pour chaque paramètre, de sorte qu'un fichier de configuration ne soit nécessaire que pour outrepasser ces valeurs par défaut lorsqu'elles sont inappropriées dans une installation particulière. Et donc, l'exigence de configurabilité est une assurance qu'il sera POSSIBLE d'outrepasser la valeur par défaut quand nécessaire, même dans un produit seulement binaire ou fondé sur une mémoire en lecture seule.

Le présent document exige une valeur particulière par défaut dans certains cas. Le choix des valeurs par défaut est une question sensible lorsque l'élément de configuration contrôle l'adaptation aux systèmes fautifs existants. Si l'Internet doit réussir à converger pour réaliser l'interopérabilité, les valeurs par défaut incorporées dans les mises en œuvre doivent mettre en œuvre le protocole officiel, et non les "dis-configurations" pour s'accommoder des mises en œuvre fautives. Bien que des considérations commerciales aient conduit certains fabricants à choisir des valeurs par défaut dis-configurées, nous invitons les fabricants à choisir des valeurs par défauts conformes à la présente norme.

Finalement, on note que le fabricant a besoin de fournir une documentation adéquate sur tous les paramètres de configuration, leurs limites et leurs effets.

1.3 Lecture de ce document

1.3.1 Organisation

La mise en couche de protocoles, qui est généralement utilisée comme principe d'organisation pour la mise en œuvre de logiciels réseau, a aussi été utilisée pour organiser le présent document. Pour décrire les règles, on suppose une mise en œuvre qui reflète strictement la mise en couche des protocoles. Et donc, les trois sections majeures suivantes spécifient les exigences pour, respectivement, la couche de liaison, la couche internet, et la couche transport. Une RFC sœur [RFC1123] couvre les logiciels de niveau application. Cette organisation en couches a été choisie pour sa simplicité et sa clarté.

Cependant, une mise en couche stricte serait un modèle imparfait, à la fois pour la suite des protocoles et pour les approches de mise en œuvre recommandées. Les protocoles des différentes couches interagissent de façons complexes parfois subtiles, et des fonctions particulières impliquent souvent plusieurs couches. Il y a de nombreux choix de conception dans une mise en œuvre, dont beaucoup impliquent une "rupture" créative de la mise en couche stricte.

Le présent document décrit les interfaces de service conceptuelles entre les couches en utilisant une notation fonctionnelle ("appel de procédure"), comme celle utilisée dans la spécification TCP [RFC0793]. Une mise en œuvre d'hôte doit prendre en charge les flux d'informations logiques impliqués par ces appels, mais n'a pas littéralement besoin de mettre en œuvre les appels eux-mêmes. Par exemple, de nombreuses mises en œuvre reflètent le couplage entre la couche transport et la couche IP en leur donnant un accès partagé aux structures de données communes. Ces structures de données, plutôt que des appels de procédure explicites, sont alors l'agence de passage de la plupart des informations nécessaires.

En général, chaque section majeure du présent document est organisée avec les paragraphes suivants :

(1) Introduction

(2) Survol du protocole -- considère les documents de spécification de protocole section par section, en corrigeant les erreurs, établissant les exigences qui pourraient être ambiguës ou mal définies, et en fournissant des éclaircissements ou des explications.

(3) Questions spécifiques – expose les questions de conception et de mise en œuvre du protocole qui ne sont pas incluses dans le survol du protocole.

(4) Interfaces – expose l'interface de service avec la prochaine couche supérieure.

(5) Résumé -- contient un résumé des exigences de la section.

Sur de nombreux sujets individuels du présent document, il y a des paragraphes entre parenthèses marqués "Discussion" ou"Mise en œuvre". Ces paragraphes sont destinés à apporter des éclaircissements et des explications au texte de l'exigence qui les précède. Ils comportent aussi des suggestions sur des directions ou développements futurs possibles. Les éléments de mise en œuvre contiennent des suggestions d'approches qu'un développeur pourrait vouloir prendre en considération.

Les paragraphe de résumé sont destinés à servir de guides et d'index pour le texte, mais sont nécessairement elliptiques et incomplets. Les résumés ne devraient jamais être utilisés ou référencés séparément de la RFC complète.

1.3.2 Exigences

Dans le présent document, les mots qui sont utilisés pour définir la signification de chaque exigence particulière sont en majuscules.

Ces mots sont :

* "DOIT"
  Ce mot ou l'adjectif "EXIGÉ" signifie que l'élément est un exigence absolue de la spécification.

* "DEVRAIT"
  Ce mot ou l'adjectif "RECOMMANDÉ" signifie qu'il peut exister des raisons valides dans des circonstances particulières pour ignorer cet élément, mais les pleines implications devraient en être comprises et le cas soigneusement soupesé avant de choisir une voie différente.

* "PEUT"
  Ce mot ou l'adjectif "FACULTATIF" signifie que cet élément est vraiment facultatif. Un fabricant peut choisir d'inclure l'élément parce que par exemple, un marché particulier l'exige ou parce qu'il améliore le produit ; un autre fabricant peut omettre le même élément.

Une mise en œuvre n'est pas conforme si elle échoue à satisfaire à une ou plusieurs des exigences marquées DOIT pour les protocoles qu'elle met en œuvre. Une mise en œuvre qui satisfait à toutes les exigences marquées DOIT et à toutes celles marquées DEVRAIT pour ses protocoles est dite être "inconditionnellement conforme" ; celle qui satisfait à tous les DOIT mais pas à tous les DEVRAIT pour ses protocoles est dite "conditionnellement conforme".

1.3.3 Terminologie

Le présent document utilise les termes techniques suivants :

Segment : Un segment est l'unité de transmission de bout en bout dans le protocole TCP. Un segment consiste en un en-tête TCP suivi des données d'application. Un segment est transmis par encapsulation au sein d'un datagramme IP.

Message : Dans cette description des protocoles de la couche inférieure, un message est l'unité de transmission dans un protocole de couche transport. En particulier, un segment TCP est un message. Un message consiste en un en-tête de protocole de transport suivi par des données de protocole d'application. Pour être transmis de bout en bout à travers l'Internet, un message doit être encapsulé à l'intérieur d'un datagramme.

Datagramme IP : Un datagramme IP est l'unité de transmission de bout en bout dans le protocole IP. Un datagramme IP consiste en un en-tête IP suivi par les données de couche transport, c'est-à-dire, d'un en-tête IP suivi par un message. Dans la description de la couche internet (Section 3), le terme de "datagramme" non qualifié devrait être compris comme se référant à un datagramme IP.

Paquet : Un paquet est l'unité de données passée à travers l'interface entre la couche internet et la couche de liaison. Il inclut un en-tête IP et des données. Un paquet peut être un datagramme IP complet ou un fragment d'un datagramme IP.

Trame : Une trame est l'unité de transmission dans un protocole de couche de liaison, et consiste en un en-tête de couche de liaison suivi par un paquet.

Réseau connecté : Le réseau auquel un hôte est interfacé est souvent appelé le "réseau local" ou le "sous-réseau" par rapport à cet hôte. Cependant, ces termes peuvent causer une certaine confusion, et nous utiliserons donc le terme "réseau connecté" dans le présent document.

Multi-rattachement : Un hôte est dit être à multi-rattachement ou à rattachement multiple si il a plusieurs adresses IP. Pour un exposé sur le multi-rattachement, voir au paragraphe 3.3.4 ci-dessous.

Interface de réseau physique : C'est une interface physique à un réseau connecté et qui a une (éventuellement unique) adresse de couche de liaison. Plusieurs interfaces de réseau physique sur un seul hôte peuvent partager la même adresse de couche de liaison, mais l'adresse doit être unique pour les différents hôtes sur le même réseau physique.

Interface [réseau] logique : On définit une interface [réseau] logique comme un chemin logique, distingué par une adresse IP unique, vers un réseau connecté. Voir le paragraphe 3.3.4.

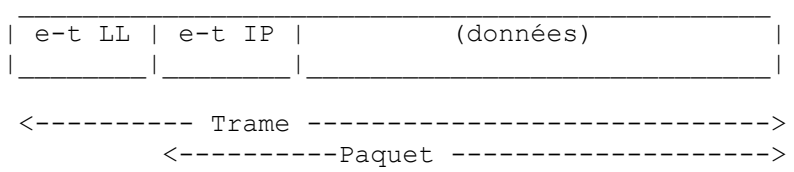
Adresse spécifique de destination : C'est l'adresse de destination effective d'un datagramme, même si il est en diffusion ou en diffusion groupée ; voir au paragraphe 3.2.1.3.

Chemin : À un moment donné, tous les datagrammes IP provenant d'un hôte de source particulier pour un hôte de destination particulier traverseront normalement la même séquence de routeurs. On utilise le terme de "chemin" pour cette séquence. Noter qu'un chemin est unidirectionnel ; il n'est pas inhabituel d'avoir des chemins différents dans les deux directions entre une paire d'hôtes donnée.

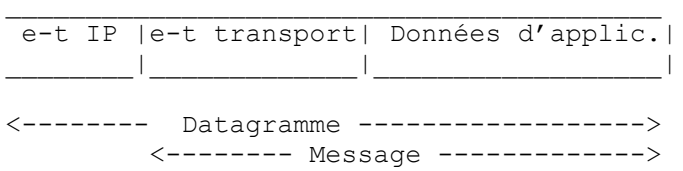
MTU : Unité de transmission maximum ; c'est-à-dire, la taille du plus grand paquet qui peut être transmis.

Les termes trame, paquet, datagramme, message, et segment sont illustrés par les diagrammes schématiques suivants :

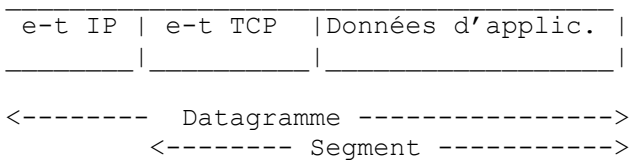
A. Transmission sur réseau connecté : *(e-t = en-tête)*



B. Avant fragmentation IP ou après réassemblage IP :



ou, pour TCP :



1.4 Remerciements

Le présent document incorpore des contributions et commentaires provenant d'un large groupe d'experts du protocole Internet, comprenant des représentants des universités et laboratoires de recherche, de fabricants, et d'agences gouvernementales. Il a été principalement confectionné par le groupe de travail Exigences pour les hôtes de l'Équipe d'ingénierie de l'Internet (IETF).

L'éditeur remercie tout spécialement le dévouement infatigable des personnes suivantes, qui ont participé à de nombreuses et longues réunions et généré 3 millions d'octets de messagerie électronique pendant les 18 derniers mois d'élaboration de ce document : Philip Almquist, Dave Borman (Cray Research), Noel Chiappa, Dave Crocker (DEC), Steve Deering (Stanford), Mike Karels (Berkeley), Phil Karn (Bellcore), John Lekashman (NASA), Charles Lynn (BBN), Keith McCloghrie (TWG), Paul Mockapetris (ISI), Thomas Narten (Purdue), Craig Partridge (BBN), Drew Perkins (CMU), et James Van Bokkelen (FTP Software).

De plus, les personnes suivantes ont apporté des contributions majeures à l'effort commun : Bill Barns (Mitre), Steve Bellovin (AT&T), Mike Brescia (BBN), Ed Cain (DCA), Annette DeSchon (ISI), Martin Gross (DCA), Phill Gross (NRI), Charles Hedrick (Rutgers), Van Jacobson (LBL), John Klensin (MIT), Mark Lottor (SRI), Milo Medin (NASA), Bill Melohn (Sun Microsystems), Greg Minshall (Kinetics), Jeff Mogul (DEC), John Mullen (CMC), Jon Postel (ISI), John Romkey (Epilogue Technology), et Mike StJohns (DCA).

Les personnes suivantes ont aussi apporté des contributions significatives dans des domaines particuliers : Eric Allman (Berkeley), Rob Austein (MIT), Art Berggreen (ACC), Keith Bostic (Berkeley), Vint Cerf (NRI), Wayne Hathaway (NASA), Matt Korn (IBM), Erik Naggum (Naggum Software, Norway), Robert Ullmann (Prime Computer), David Waitzman (BBN), Frank Wancho (USA), Arun Welch (Ohio State), Bill Westfield (Cisco), et Rayan Zachariassen (Toronto).

Nos remerciements à tous, y compris à tous les contributeurs qui ont été malencontreusement omis de cette liste.

2. Couche de liaison

2.1 Introduction

Tous les systèmes Internet, aussi bien hôtes que routeurs, ont les mêmes exigences pour les protocoles de couche de liaison des données. Ces exigences sont données au Chapitre 3 de "Exigences pour les routeurs Internet" [RFC1009], augmentées des matériaux de la présente section.

2.2 Survol du protocole

Rien

2.3 Questions spécifiques

2.3.1 Négociation du protocole d'en-queue

Le protocole d'en-queue [RFC0893] pour l'encapsulation de couche de liaison PEUT être utilisé, mais seulement lorsqu'on a vérifié que les deux systèmes (hôte ou routeur) impliqués dans la communication de couche de liaison mettent en œuvre des en-queues. Si le système ne négocie pas de façon dynamique l'utilisation du protocole d'en-queue destination par destination, la configuration par défaut DOIT désactiver le protocole.

Discussion :
Le protocole d'en-queue est une technique d'encapsulation de couche de liaison qui réarrange le contenu de données des paquets envoyés sur le réseau physique. Dans certains cas, les en-queues améliorent le débit des protocoles de couche supérieure en réduisant la quantité de données en les copiant au sein du système d'exploitation. Les protocoles de couche supérieure ne sont pas informés de l'utilisation des en-queues, mais l'hôte expéditeur comme l'hôte receveur DOIVENT comprendre le protocole si il est utilisé.

Une utilisation inappropriée des en-queues peut déboucher sur des symptômes très ennuyeux. Seuls les paquets ayant des attributs de taille spécifiques sont encapsulés à l'aide d'en-queues, et normalement seule une petite fraction des paquets échangés ont ces attributs. Et donc, si un système qui utilise des en-queues échange des paquets avec un système qui ne le fait pas, certains paquets disparaissent dans un trou noir alors que d'autres sont bien livrés.

Mise en œuvre :
Sur un Ethernet, les paquets encapsulés avec des en-queues utilisent un type Ethernet distinct [RFC0893], et la négociation d'en-queue est effectuée au moment où ARP est utilisé pour découvrir l'adresse de couche de liaison d'un système de destination.

Spécifiquement, l'échange ARP est mené à bien de la façon usuelle en utilisant le type normal de protocole IP, mais un hôte qui veut utiliser des en-queues enverra un paquet "réponse ARP d'en-queue" supplémentaire, c'est-à-dire, une réponse ARP qui spécifie le type de protocole d'encapsulation d'en-queue mais a par ailleurs le format d'une réponse ARP normale. Si un hôte configuré pour utiliser des en-queues reçoit un message de réponse ARP d'en-queue d'une machine distante, il peut ajouter cette machine à la liste des machines qui comprennent les en-queues, par exemple, en marquant l'entrée correspondante dans l'antémémoire ARP.

Les hôtes qui souhaitent recevoir des encapsulations d'en-queue envoient des réponse ARP d'en-queue chaque fois qu'ils achèvent des échanges de messages ARP normaux pour IP. Et donc, un hôte qui a reçu une demande ARP pour son adresse de protocole IP enverra une réponse ARP d'en-queue en plus de la réponse ARP IP normale ; un hôte qui envoie la demande ARP IP enverra une réponse ARP d'en-queue lorsqu'il a reçu la réponse ARP IP correspondante. De cette façon, l'hôte demandeur ou l'hôte qui répond dans un échange ARP IP peut demander à recevoir des encapsulations d'en-queue.

Ce schéma, qui utilise des paquets de réponse ARP d'en-queue supplémentaires plutôt que d'envoyer une demande ARP de type de protocole d'en-queue, a été conçu pour éviter un échange continuel de paquets ARP avec un hôte qui se comporte mal, et qui, contrairement à toutes les spécifications ou au bon sens, a répondu à une réponse ARP d'en-queue par une autre réponse ARP pour IP. Ce problème n'est évité par l'envoi d'une réponse ARP d'en-queue en réponse à une réponse ARP IP que lorsque la réponse ARP IP répond à une demande en cours ; ceci est vrai lorsque l'adresse matérielle pour l'hôte n'est toujours pas connue lors de la réception de la réponse ARP IP. Une réponse ARP d'en-queue peut toujours être envoyée avec une réponse ARP IP qui répond à une demande ARP IP.

2.3.2 Protocole de résolution d'adresse -- ARP

2.3.2.1 Validation de mémoire cache ARP

Une mise en œuvre du protocole de résolution d'adresse (ARP) [RFC0826] DOIT fournir un mécanisme de purge des entrées d'antémémoire périmées. Si ce mécanisme implique une temporisation, il DEVRAIT être possible de configurer la valeur de la temporisation.

Un mécanisme empêchant le débordement d'ARP (par l'envoi répétitif d'une demande ARP à la même adresse IP, à haut débit) DOIT être inclus. Le débit maximum recommandé est de une par seconde par destination.

Discussion :
La spécification ARP [RFC0826] suggère mais n'exige pas un mécanisme de temporisation pour invalider les entrées de mémoire cache lorsque les hôtes changent leurs adresses Ethernet. La prévalence des mandataires ARP (voir au paragraphe 2.4 de la [RFC1009]) a augmenté de façon significative la probabilité que des entrées d'antémémoire deviennent invalides dans les hôtes, et donc, un mécanisme d'invalidation d'antémémoire ARP est maintenant nécessaire pour les hôtes. Même en l'absence de mandataire ARP, une temporisation d'antémémoire de longue durée est utile afin de corriger automatiquement de mauvaises données ARP qui auraient pu être mises en antémémoire.

Mise en œuvre :
Quatre mécanismes ont été utilisés, parfois en combinaison, pour purger les entrées d'antémémoire périmées.

(1) Temporisation – Amener périodiquement les entrées d'antémémoire en fin de temporisation, même si elles sont utilisées. Noter que cette temporisation devrait être redémarrée lorsque l'entrée d'antémémoire est "rafraîchie" (en observant les champs de source, sans considération de l'adresse cible, sur la diffusion ARP provenant du système en question). Pour les situations de mandataire ARP, la temporisation doit être de l'ordre de la minute.

(2) Interrogation d'envoi individuel – Questionner activement l'hôte distant en lui envoyant périodiquement une demande ARP en point à point, et supprimer l'entrée si aucune réponse ARP n'est reçue sur N interrogations successives. Là encore, la temporisation devrait être de l'ordre de la minute, et normalement N est 2.

(3) Avis de couche de liaison – Si le pilote de couche de liaison détecte un problème de livraison, vider l'entrée d'antémémoire ARP correspondante.

(4) Avis de couche supérieure – Fournit un appel de la couche Internet à la couche de liaison pour indiquer un problème de livraison. L'effet de cet appel sera d'invalider l'entrée d'antémémoire correspondante. Cet appel serait analogue à l'appel "ADVISE_DELIVPROB()" de la couche transport à la couche Internet (voir au paragraphe 3.4), et en fait, la routine ADVISE_DELIVPROB pourrait à son tour appeler la routine d'avis de couche de liaison pour invalider l'entrée d'antémémoire ARP.

Les approches (1) et (2) impliquent des fins de temporisation d'antémémoire ARP de l'ordre d'une minute ou moins. En l'absence de mandataire ARP, une temporisation aussi courte pourrait créer une redondance de trafic notable sur un très gros Ethernet. Donc, il peut être nécessaire de configurer un hôte pour allonger la temporisation d'antémémoire ARP.

2.3.2.2 File d'attente de paquet ARP

La couche de liaison DEVRAIT sauvegarder (plutôt qu'éliminer) au moins un paquet (le dernier) de chaque ensemble de paquets destinés à la même adresse IP non résolue, et transmettre le paquet sauvegardé lorsque l'adresse a été résolue.

Discussion :
Ne pas suivre cette recommandation cause la perte du premier paquet de chaque échange. Bien que les protocoles de couche supérieure puissent généralement s'arranger de la perte de paquet par la retransmission, la perte de paquet a un impact sur les performances. Par exemple, la perte d'une demande d'ouverture TCP cause une surévaluation de l'estimation initiale du délai d'aller-retour. Les applications fondées sur UDP telles que le système des noms de domaine sont plus sérieusement affectées.

2.3.3 Ethernet et encapsulation IEEE 802

L'encapsulation IP pour les Ethernets est décrite dans la [RFC0894], alors que la [RFC1042] décrit l'encapsulation IP pour les réseaux IEEE 802. La RFC-1042 enrichit et remplace l'exposé du paragraphe 3.4 de la [RFC1009].

Chaque hôte internet connecté à un câble Ethernet à 10 Mbit/s :
- DOIT être capable d'envoyer et recevoir des paquets en utilisant l'encapsulation de la RFC-894 ;
- DEVRAIT être capable de recevoir des paquets de la RFC-1042, entremêlés à des paquets de la RFC-894 ;
- PEUT être capable d'envoyer des paquets en utilisant l'encapsulation de la RFC-1042.

Un hôte Internet qui met en œuvre l'envoi d'encapsulations à la fois de la RFC-894 et de la RFC-1042 DOIT fournir une commutation de configuration pour choisir celle qui est envoyée, et cette commutation DOIT être par défaut celle de la RFC-894.

Noter que l'encapsulation IP standard dans la RFC-1042 n'utilise pas la valeur d'identifiant de protocole (K1=6) que l'IEEE a réservé pour IP ; elle utilise à la place une valeur (K1=170) qui implique une extension (le "SNAP") qui peut être utilisée pour contenir le champ Ether-Type. Un système Internet NE DOIT PAS envoyer de paquets 802 en utilisant K1=6.

La traduction d'adresse des adresses Internet en adresses de couche de liaison sur les réseaux Ethernet et IEEE 802 DOIT être gérée par le protocole de résolution d'adresse (ARP).

La MTU est de 1500 pour un Ethernet et de 1492 pour 802.3.

Discussion :
La spécification IEEE 802.3 prévoit le fonctionnement sur un câble Ethernet à 10 Mbit/s, auquel cas les trames Ethernet et IEEE 802.3 peuvent être physiquement mélangées. Un receveur peut distinguer les trames Ethernet et 802.3 par la valeur du champ Longueur de 802.3 ; ce champ de deux octets coïncide dans l'en-tête avec le champ Ether-Type d'une trame Ethernet. En particulier, le champ Longueur de 802.3 doit être inférieur ou égal à 1500, alors que toutes les valeurs d'Ether-Type valides sont supérieures à 1500.

Un autre problème de compatibilité survient avec les diffusions de couche de liaison. Une diffusion envoyée avec un tramage ne sera pas vue par les hôtes qui ne peuvent recevoir que l'autre tramage.

Les dispositions de ce paragraphe ont été conçues pour fournir une interopération directe entre les systèmes à capacité 894 et ceux à capacité 1042 sur le même câble, dans la mesure maximum possible. Il est prévu de prendre en charge la présente situation lorsque les systèmes 894 seuls prédominent, tout en permettant une transition facile dans un avenir possible où les systèmes à capacité 1042 deviendraient courants.

Noter que les systèmes à capacité 894 seule ne peuvent pas interopérer directement avec les systèmes à capacité 1042 seule. Si les deux types de systèmes sont établis sur deux réseaux logiques différents sur le même câble, ils ne peuvent communiquer qu'à travers une passerelle IP. De plus, il n'est pas utile ou même possible à un hôte à double format de découvrir automatiquement quel format envoyer, à cause du problème des diffusions de couche de liaison.

2.4 Interface de couche Liaison/Internet

L'interface de réception de paquet entre la couche IP et la couche de liaison DOIT inclure un fanion pour indiquer si le paquet entrant est adressé à une adresse de diffusion de couche de liaison.

Discussion :
Bien que la couche IP ne sache généralement pas les adresses de couche de liaison (car chaque support de réseau différent a normalement un format d'adresse différent), l'adresse de diffusion sur un support capable de diffusion est un cas particulier important. Voir au paragraphe 3.2.2, et particulièrement la discussion concernant les tempêtes de diffusion.

L'interface d'envoi de paquet entre les couches IP et de liaison DOIT inclure le champ TOS de 5 bits (voir au paragraphe 3.2.1.6).

La couche de liaison NE DOIT PAS rapporter une erreur Destination injoignable sur IP seulement parce qu'il n'y a pas d'entrée d'antémémoire ARP pour une destination.

2.5 Résumé des exigences de couche de liaison

Caractéritique	Parag.	DOIT	DEVRAIT	PEUT	NE DEVRAIT PAS	NE DOIT PAS
Encapsulation d'en-queue	2.3.1		x			
Envoie les en-queue par défaut sans négociation	2.3.1			x		
ARP	2.3.2					
Purge des entrées d'antémémoire ARP périmées	2.3.2.1	x				
Empêche les débordements d'ARP	2.3.2.1	x				
Temporisation configurable d'antémémoire	2.3.2.1		x			
Sauvegarde au moins un paquet non résolu (dernier)	2.3.2.2		x			
Encapsulation Ethernet et IEEE 802	2.3.3					
Hôte capable de :	2.3.3					

Envoyer et recevoir l'encapsulation de RFC 894	2.3.3	x				
Recevoir l'encapsulation de RFC-1042	2.3.3		x			
Envoyer l'encapsulation de RFC-1042	2.3.3		x			
Puis configurer sw. pour choisir la RFC-894 par défaut	2.3.3	x				
Envoyer l'encapsulation K1=6	2.3.3					x
Utiliser ARP sur les réseaux Ethernet et IEEE 802	2.3.3	x				
Diff. du rapport de couche de liaison à la couche IP	2.4	x				
La couche IP passe le TOS à la couche de liaison	2.4	x				
Aucune entrée d'antémémoire IP n'est traitée comme destination injoignable	2.4					x

3 Protocoles de couche Internet

3.1 Introduction

Le principe de robustesse : "Soyez libéraux dans ce que vous acceptez, et conservateur dans ce que vous envoyez" est particulièrement important à la couche Internet, où un hôte au mauvais comportement peut dénier le service de l'Internet à de nombreux autres hôtes.

Les normes de protocole utilisées à la couche Internet sont :

- La [RFC0791] qui définit le protocole IP et sert d'introduction à l'architecture de l'Internet.
- La [RFC0792] qui définit ICMP, qui fournit les fonctions de routage, de diagnostic et d'erreur pour IP. Bien que les messages ICMP soient encapsulés dans les datagrammes IP, le traitement ICMP est considéré comme faisant partie de la couche IP (et il est normalement mis en œuvre à ce titre). Voir au paragraphe 3.2.2.
- La [RFC0950] qui définit l'extension de sous-réseau obligatoire pour l'architecture d'adressage.
- La [RFC1112] qui définit le protocole de gestion de groupe Internet (IGMP), au titre d'une extension recommandée des hôtes et des interfaces hôte-passerelle pour la prise en charge de la diffusion groupée à l'échelle de l'Internet au niveau de la couche IP. Voir au paragraphe 3.2.3.

La cible d'une diffusion groupée IP peut être un groupe arbitraire d'hôtes Internet. La diffusion groupée IP est conçue comme une extension naturelle du dispositif de diffusion groupée de couche de liaison de certains réseaux, et elle procure un moyen normalisé pour l'accès local à de telles facilités de diffusion groupée de couche de liaison.

D'autres références importantes sont énumérées à la Section 5 du présent document.

La couche Internet du logiciel d'hôte DOIT mettre en œuvre à la fois IP et ICMP. Voir au paragraphe 3.3.7 les exigences de la prise en charge de IGMP.

La couche IP hôte a deux fonctions de base : (1) choisir le routeur ou hôte du "prochain bond" pour les datagrammes IP sortants et (2) ré assembler les datagrammes IP entrants. La couche IP peut aussi (3) mettre en œuvre une fragmentation intentionnelle des datagrammes sortants. Finalement, la couche IP doit (4) fournir une fonction de diagnostic et de recherche d'erreur. On prévoit que les fonctions de la couche IP augmentent un peu à l'avenir, avec le développement d'autres facilités de commandes et de gestion Internet.

Pour les datagrammes normaux, le traitement est direct. Pour les datagrammes entrants, la couche IP :

(1) vérifie que le datagramme est correctement formaté ;
(2) vérifie qu'il est destiné à l'hôte local ;
(3) traite les options ;
(4) ré assemble le datagramme si nécessaire ; et
(5) passe le message encapsulé au module approprié de protocole de couche transport.

Pour les datagrammes sortants, la couche IP :

(1) établit tous les champs qui ne l'ont pas été par la couche transport ;
(2) choisit le premier bond correct sur le réseau connecté (un processus appelé "acheminement");
(3) fragmente le datagramme si nécessaire et si la fragmentation intentionnelle est mise en œuvre (voir au paragraphe 3.3.3) ; et
(4) passe le ou les paquets au pilote de couche liaison approprié.

Un hôte est dit à multi rattachement si il a plusieurs adresses IP. Le multi rattachement introduit une confusion et une complexité considérable dans la suite de protocoles, et c'est un domaine dans lequel l'architecture Internet est bien loin de résoudre tous les problèmes. Il y a deux domaines distincts de problèmes du multi rattachement :
(1) Multi rattachement local -- l'hôte lui-même est à multi rattachement ; ou
(2) Multi rattachement distant – l'hôte local a besoin de communiquer avec un hôte à multi rattachement distant.

À présent, le multi rattachement distant DOIT être traité à la couche d'application, comme exposé dans la RFC sœur [RFC1123]. Un hôte PEUT prendre en charge le multi rattachement local, qui est exposé dans le présent document, et en particulier au paragraphe 3.3.4.

Tout hôte qui transmet des datagrammes générés par un autre hôte agit comme un routeur et DOIT aussi satisfaire aux spécifications traitant des exigences des passerelles [RFC1009]. Un hôte Internet qui comporte un code de routeur incorporé DOIT avoir une commutation de configuration pour désactiver la fonction routeur, et ce commutateur DOIT par défaut revenir au mode non-routeur. Dans ce mode, un datagramme arrivant à travers une interface ne sera pas transmis à un autre hôte ou routeur (sauf si c'est un acheminement de source) sans considération de savoir si l'hôte est à rattachement unique ou multiple. Le logiciel d'hôte NE DOIT PAS automatiquement passer en mode routeur si l'hôte a plus d'une interface, car l'opérateur de la machine peut ne vouloir ni fournir ce service ni être compétent pour le faire.

Dans la suite du texte, l'action spécifiée dans certains cas est de "éliminer en silence" un datagramme reçu. Cela signifie que le datagramme sera éliminé sans autre traitement et que l'hôte n'enverra aucun message d'erreur ICMP (voir au paragraphe 3.2.2) à ce titre. Cependant, pour des raisons de diagnostic des problèmes, un hôte DEVRAIT fournir la capacité d'enregistrer l'erreur (voir au paragraphe 1.2.3) y compris le contenu du datagramme éliminé en silence, et DEVRAIT enregistrer l'événement dans un compteur statistique.

Discussion :
L'élimination silencieuse de datagrammes erronés est généralement destinés à empêcher les "tempêtes de diffusion".

3.2 Survol du protocole

3.2.1 Protocole Internet -- IP

3.2.1.1 Numéro de version : RFC-791 paragraphe 3.1

Un datagramme dont le numéro de version n'est pas 4 DOIT être éliminé en silence.

3.2.1.2 Somme de contrôle : RFC-791 paragraphe 3.1

Un hôte DOIT vérifier la somme de contrôle d'en-tête IP sur chaque datagramme reçu et éliminer en silence tout datagramme qui a une mauvaise somme de contrôle.

3.2.1.3 Adressage : RFC-791 paragraphe 3.2

Il y a maintenant cinq classes d'adresses IP : de la classe A à la classe E. Les adresses de classe D sont utilisées pour la diffusion groupée IP [RFC1112], alors que les adresses de classe E sont réservées pour une utilisation expérimentale.

Une adresse de diffusion groupée (Classe D) est une adresse logique de 28 bits qui vaut pour un groupe d'hôtes, et peut être soit permanente soit transitoire. Les adresses de diffusion groupée permanentes sont allouées par l'Autorité d'allocation des numéros de l'Internet [RFC1700], alors que les adresses transitoires peuvent être allouées de façon dynamique à des groupes temporaires. L'adhésion aux groupes est déterminée de façon dynamique à l'aide de IGMP [RFC1112].

Nous allons maintenant résumer les cas particuliers importants pour les adresses IP de classe A, B, et C, en utilisant la notation suivante pour une adresse IP :

{ <Numéro-de-réseau>, <Numéro-d'hôte> }
ou
{ <Numéro-de-réseau>, <Numéro-de-sous-réseau>, <Numéro-d'hôte> }

et la notation "-1" pour un champ qui contient tous ses bits à 1. Cette notation n'est pas destinée à impliquer que les bits 1 dans un gabarit d'adresse doivent être contigus.

(a) { 0, 0 }
Cet hôte sur ce réseau. NE DOIT PAS être envoyé, excepté comme adresse de source au titre d'une procédure d'initialisation par laquelle l'hôte apprend sa propre adresse IP.

Voir aussi au paragraphe 3.3.6 une utilisation non normalisée de {0,0}.

(b) { 0, <Numéro-d'hôte> }
L'hôte spécifié sur ce réseau. NE DOIT PAS être envoyé, excepté comme adresse de source au titre d'une procédure d'initialisation par laquelle l'hôte apprend son adresse IP complète.

(c) { -1, -1 }
Diffusion limitée . NE DOIT PAS être utilisé comme adresse de source.

Un datagramme avec cette adresse de destination sera reçu par tous les hôtes sur le réseau physique connecté mais ne sera pas transmis en-dehors de ce réseau.

(d) { <Numéro-de-réseau>, -1 }
Diffusion dirigée sur le réseau spécifié. NE DOIT PAS être utilisé comme adresse de source.

(e) { <Numéro-de-réseau>, <Numéro-de-sous-réseau>, -1 }
Diffusion dirigée sur le sous-réseau spécifié. NE DOIT PAS être utilisé comme adresse de source.

(f) { <Numéro-de-réseau>, -1, -1 }
Diffusion dirigée sur tous les sous-réseaux du réseau à sous-réseaux spécifié. NE DOIT PAS être utilisé comme adresse de source.

(g) { 127, <any> }
Adresse de bouclage arrière d'hôte interne. Les adresses de cette forme NE DOIVENT PAS apparaître en-dehors d'un hôte.

Le <Numéro-de-réseau> est alloué administrativement de telle sorte que sa valeur soit unique au monde.

Les adresses IP ne sont pas autorisées à avoir la valeur 0 ou -1 pour tout champ <Numéro-d'hôte>, <Numéro-de-réseau>, ou <Numéro-de-sous-réseau> (excepté dans les cas particuliers dont la liste figure ci-dessus). Cela implique que chacun de des champs aura au moins deux octets de long.

Pour des précisions sur les adresses de diffusion, voir au paragraphe 3.3.6.

Un hôte DOIT prendre en charge les extensions de sous-réseau à IP [RFC0950]. Il en résulte qu'il y aura un gabarit d'adresse de la forme : {-1, -1, 0} associé à chaque adresse IP locale de l'hôte ; voir aux paragraphes 3.2.2.9 et 3.3.1.1.

Lorsque un hôte envoie un datagramme, l'adresse IP de source DOIT être une de ses propres adresses IP (mais pas une adresse de diffusion ou diffusion groupée).

Un hôte DOIT éliminer en silence un datagramme entrant qui n'est pas destiné à l'hôte. Un datagramme entrant est destiné à l'hôte si le champ adresse de destination du datagramme est : (1) une (des) adresse(s) IP de l'hôte ; ou (2) une adresse de diffusion IP valide pour le réseau connecté ; ou (3) l'adresse d'un groupe de diffusion dont l'hôte est un membre sur l'interface physique entrante.

Pour la plupart des besoins, un datagramme adressé à une destination de diffusion ou de diffusion groupée est traité comme si il avait été adressé à une des adresses IP de l'hôte ; on utilise le terme "adresse de destination spécifique " pour l'adresse IP locale équivalente de l'hôte. L'adresse de destination spécifique est définie comme étant l'adresse de destination dans l'en-tête IP sauf si l'en-tête contient une adresse de diffusion ou de diffusion groupée, auquel cas la destination spécifique est une adresse IP allouée à l'interface physique sur laquelle le datagramme est arrivé.

Un hôte DOIT éliminer en silence un datagramme entrant contenant une adresse IP de source invalide selon les règles de la présente section. Cette validation pourrait être faite dans la couche IP ou par chaque protocole dans la couche transport.

Discussion : Un datagramme mal adressé peut être causé par la diffusion à la couche de liaison d'un datagramme en envoi individuel ou par un routeur ou hôte confus ou mal configuré.

Un objectif architectural des hôtes Internet était de permettre que les adresses IP soient des nombres de 32 bits sans caractéristique, évitant les algorithmes qui exigent une connaissance du format d'adresse IP. Autrement, tout futur changement du format ou de l'interprétation des adresses IP exigerait des changements du logiciel de l'hôte. Cependant, la validation des adresses de diffusion et de diffusion groupée viole cet objectif ; quelques autres violations sont décrites ailleurs dans le présent document.

Les développeurs devraient savoir que les applications qui dépendent des adresses de diffusion dirigées sur tous les sous-réseaux (f) peuvent être inutilisables sur certains réseaux. La diffusion à tous les sous-réseaux n'est pas très largement mise en œuvre à présent dans les routeurs des fabricants, et même lorsqu'elle est mise en œuvre, une administration de réseau particulière peut la désactiver dans la configuration du routeur.

3.2.1.4 Fragmentation et réassemblage : RFC-791 paragraphe 3.2

Le modèle Internet exige que chaque hôte prenne en charge le réassemblage. Voir aux paragraphes 3.3.2 et 3.3.3 les exigences sur la fragmentation et le réassemblage.

3.2.1.5 Identification : RFC-791 paragraphe 3.2

Lors de l'envoi d'une copie identique d'un datagramme précédent, un hôte PEUT facultativement conserver le même champ Identification dans la copie.

Discussion :
Certains experts en protocole Internet ont soutenu que lorsque un hôte envoie une copie identique d'un datagramme précédent, la nouvelle copie devrait contenir la même valeur Identification que l'original. Deux avantages sont suggérés : (1) si les datagrammes sont fragmentés et si certains des fragments sont perdus, le receveur peut être capable de reconstruire un datagramme complet à partir des fragments de l'original et des copies ; (2) un routeur encombré peut utiliser le champ Identification IP (et Décalage de fragment) pour éliminer les datagrammes dupliqués de la file d'attente.

Cependant, le schéma observé des pertes de datagramme dans l'Internet n'est pas en faveur de la probabilité que des fragments retransmis remplissent les trous du réassemblage, alors que d'autres mécanismes (par exemple, la remise en paquet de TCP sur retransmission) tendent à empêcher la retransmission d'un datagramme identique [IP:9]. Donc, nous pensons que retransmettre le même champ Identification n'est pas utile. De plus, un protocole de transport sans connexion comme UDP exigerait la coopération des programmes d'application pour conserver la même valeur d'Identification dans des datagrammes identiques.

3.2.1.6 Type-de-Service : RFC-791 paragraphe 3.2

L'octet "Type-de-Service" dans l'en-tête IP est divisé en deux sections : le champ Préséance (3 bits de plus fort poids), et un champ qui est traditionnellement appelé "Type-de-Service" ou "TOS" (5 bits de moindre poids). Dans le présent document, toutes les références à "TOS" ou au "champ TOS" se rapportent seulement aux 5 bits de moindre poids.

Le champ Préséance est destiné aux applications de protocole Internet du Département de la Défense. L'utilisation de valeurs différentes de zéro dans ce champ sort du domaine d'application du présent document et de la spécification de la norme IP. Les fabricants devraient consulter l'agence de communication de la Défense (DCA) pour des lignes directrices sur l'utilisation du champ Préséance et ses implications pour les autres couches de protocole. Cependant, les fabricants devraient noter que l'utilisation de Préséance va très vraisemblablement exiger que sa valeur soit communiquée entre les couches de protocole de la même façon qu'est communiqué le champ TOS.

La couche IP DOIT fournir le moyen que la couche transport établisse le champ TOS de chaque datagramme envoyé ; la valeur par défaut est que tous les bits sont à zéro. La couche IP DEVRAIT passer les valeurs de TOS reçues jusqu'à la couche transport.

Les transpositions particulières de couche de liaison de TOS contenues dans la RFC-795 NE DEVRAIT PAS être mises en œuvre.

Discussion :
Bien que le champ TOS ait été peu utilisé dans le passé, on prévoit qu'il va jouer un rôle croissant dans un futur proche. Il est prévu d'utiliser le champ TOS pour contrôler deux aspects du fonctionnement des routeurs : les algorithmes

d'acheminement et de mise en file d'attente. Voir à la Section 2 de la [RFC1123] les exigences qui pèsent sur les programmes d'application pour spécifier les valeurs de TOS.

Le champ TOS peut aussi être transposé dans les sélecteurs de service de couche de liaison. Cela a été appliqué pour fournir un partage effectif des lignes de série par différentes classes de trafic TCP, par exemple. Cependant, les transpositions suggérées dans la RFC-795 pour les réseaux qui étaient inclus dans l'Internet depuis 1981 sont maintenant obsolètes.

3.2.1.7 Durée de vie : RFC-791 paragraphe 3.2

Un hôte NE DOIT PAS envoyer un datagramme avec une valeur de durée de vie (TTL, *Time-to-Live*) de zéro.

Un hôte NE DOIT PAS éliminer un datagramme seulement parce qu'il a été reçu avec une TTL inférieure à 2.

La couche IP DOIT fournir le moyen que la couche transport établisse le champ TTL de chaque datagramme envoyé. Lorsqu'une valeur fixe de TTL est utilisée, elle DOIT être configurable. La valeur courante suggérée sera publiée dans la RFC "Numéros alloués".

Discussion :
Le champ TTL a deux fonctions : limiter la durée de vie des segments TCP (voir la [RFC0793], p. 28), et mettre un terme aux boucles d'acheminement Internet. Bien que TTL soit un temps en secondes, il a aussi quelques attributs d'un compte de bonds, car chaque routeur est obligé de réduire le champ TTL d'au moins un.

L'intention est que l'expiration du TTL cause l'élimination d'un datagramme par un routeur mais pas par l'hôte de destination ; cependant, les hôtes qui agissent comme des routeurs en transmettant les datagrammes doivent suivre les règles des routeurs pour le TTL.

Un protocole de couche supérieure peut vouloir régler le TTL de façon à mettre en œuvre une recherche à "expansion de portée" pour une ressource Internet. C'est utilisé par certains outils de diagnostic, et est supposé être utile pour localiser le "plus proche" serveur d'une classe donnée en utilisant la diffusion groupée IP, par exemple. Un protocole de transport particulier peut aussi vouloir spécifier sa propre limite de TTL sur la durée de vie maximum de datagramme.

Une valeur fixée doit être au moins assez grande pour le "diamètre" Internet, c'est-à-dire, le plus long chemin possible. Une valeur raisonnable est d'environ deux fois le diamètre, pour permettre la continuation de la croissance de l'Internet.

3.2.1.8 Options : RFC-791 paragraphe 3.2

Il DOIT y avoir un moyen pour que la couche transport spécifie les options IP à inclure dans les datagrammes IP transmis (voir au paragraphe 3.4).

Toutes les options IP (excepté NOP ou END-OF-LIST) reçues dans les datagrammes DOIVENT être passées à la couche transport (ou au traitement ICMP lorsque le datagramme est un message ICMP). La couche IP et la couche transport DOIVENT chacune interpréter les options IP qu'elles comprennent, et ignorer les autres en silence.

Les sections ultérieures du présent document exposent la prise en charge des options IP spécifiques exigées par ICMP, TCP, et UDP.

Discussion :
Passer toutes les options IP reçues à la couche transport est une "violation de la mise en couche stricte" délibérée qui est conçue pour faciliter l'introduction à l'avenir de nouvelles options IP pertinentes pour le transport. Chaque couche doit saisir toute option pertinente pour son propre traitement et ignorer le reste. À cette fin, toute option IP excepté NOP et END-OF-LIST inclura une spécification de sa propre longueur.

Le présent document ne définit pas l'ordre dans lequel un receveur doit traiter plusieurs options dans le même en-tête IP. Les hôtes qui envoient plusieurs options doivent être avertis que cela introduit des ambiguïtés dans la signification de certaines options lorsqu'elles sont combinées avec une option source-route.

Mise en œuvre :
La couche IP ne doit pas avoir de défaillance par suite d'une longueur d'option en dehors de la gamme possible. Par exemple, des longueurs d'option erronées ont été observées qui mettent certaines mises en œuvre IP en boucle infinie.

Voici les exigences pour les options spécifiques d'IP :

(a) Option Sécurité
Certains environnements exigent l'option Sécurité dans chaque datagramme ; une telle exigence sort du domaine d'application du présent document et de la spécification de la norme IP. Noter, cependant, que les options de sécurité décrites dans les RFC-791 et RFC-1038 sont obsolètes. Pour les applications du DoD, les constructeurs devraient consulter la [RFC1108] pour des explications.

(b) Option Identifiant de flux
Cette option est obsolète; elle NE DEVRAIT PAS être envoyée, et DOIT être ignorée en silence à réception.

(c) Option Route de source *(source route)*
Un hôte DOIT prendre en charge la génération d'un Route de source et DOIT être capable d'agir comme destination finale d'un Route de source.

Si un hôte reçoit un datagramme contenant un Route de source terminé (c'est-à-dire, le pointeur pointe au-delà du dernier champ) le datagramme a atteint sa destination finale ; l'option telle que reçue (la route enregistrée) DOIT être passée jusqu'à la couche transport (ou au traitement de message ICMP). Cette route enregistrée sera inversée et utilisée pour former un Route de source de retour pour les datagrammes de réponse (voir la discussion des options IP en Section 4). Lors de la construction d'un Route de source, il DOIT être correctement formé même si la route enregistrée inclut l'hôte de source (voir le cas (B) dans la discussion ci-dessous).

Un en-tête IP contenant plus d'une option Route de source NE DOIT PAS être envoyée ; l'effet de plusieurs options Route de source sur l'acheminement est spécifique de la mise en œuvre.

Le paragraphe 3.3.5 présente les règles pour un hôte qui agit comme bond intermédiaire dans un Route de source, c'est-à-dire, qui transmet un datagramme avec un acheminement de source.

Discussion :
Si un datagramme à acheminement de source est fragmenté, chaque fragment va contenir une copie du Route de source. Comme le traitement des options IP (y compris un Route de source) doit précéder le réassemblage, le datagramme original ne sera pas réassemblé jusqu'à avoir atteint la destination finale.

Supposons qu'un datagramme à acheminement de source doit être acheminé de l'hôte S à l'hôte D via des routeurs G1, G2, ... Gn. Il y avait une ambiguïté dans la spécification sur le fait de savoir si l'option source route dans un datagramme envoyé par S devrait être (A) ou (B) :

(A): {>>G2, G3, ... Gn, D} <--- CORRECT
(B): {S, >>G2, G3, ... Gn, D} <---- FAUX
(où >> représente le pointeur). Si (A) est envoyé, le datagramme reçu à D contiendra l'option : {G1, G2, ... Gn >>}, avec S et D comme sources et adresses IP de destination. Si (B) est envoyé, le datagramme reçu à D contiendrait à nouveau S et D comme mêmes sources et adresses IP de destination, mais l'option serait : {S, G1, ...Gn >>} ; c'est-à-dire que l'hôte d'origine serait le premier bond sur la route.

(d) Option Route enregistrée *(Record Route)*
La mise en œuvre de la génération et du traitement de l'option Record Route est FACULTATIVE.

(e) Option Horodatage *(Timestamp)*
La mise en œuvre de la génération et du traitement de l'option Horodatage est FACULTATIVE. Si elle est mise en œuvre, les règles suivantes s'appliquent :

- L'hôte d'origine DOIT enregistrer un horodatage dans une option Horodatage dont les champs d'adresse Internet ne sont pas pré-spécifiés ou dont la première adresse pré-spécifiée est l'adresse de l'interface de l'hôte.
- L'hôte de destination DOIT (si possible) ajouter l'horodatage en cours à une option Horodatage avant de passer l'option à la couche transport ou ICMP pour traitement.
- Une valeur d'horodatage DOIT suivre les règles données au paragraphe 3.2.2.8 pour le message Horodatage ICMP.

3.2.2 Protocole de message de commande Internet -- ICMP

Les messages ICMP sont regroupés en deux classes.

* les messages d'erreur ICMP :
  - Destination injoignable (voir au paragraphe 3.2.2.1)
  - Rediriger (voir au paragraphe 3.2.2.2)
  - Source éteinte (voir au paragraphe 3.2.2.3)
  - Temps excédé (voir au paragraphe 3.2.2.4)
  - Problème de paramètre (voir au paragraphe 3.2.2.5)

* les messages d'interrogation ICMP :
  - Écho (voir au paragraphe 3.2.2.6)
  - Information (voir au paragraphe 3.2.2.7)
  - Horodatage (voir au paragraphe 3.2.2.8)
  - Gabarit d'adresse (voir au paragraphe 3.2.2.9)

Si un message ICMP de type inconnu est reçu, il DOIT être éliminé en silence.

Tout message d'erreur ICMP inclut l'en-tête Internet et au moins les huit premiers octets de données du datagramme qui a déclenché l'erreur ; plus de huit octets PEUVENT être envoyés ; cet en-tête et les données DOIVENT être inchangés par rapport au datagramme reçu.

Dans les cas où la couche Internet est obligée de passer un message d'erreur ICMP à la couche transport, le numéro de protocole IP DOIT être extrait de l'en-tête original et utilisé pour choisir l'entité de protocole de transport appropriée pour traiter l'erreur.

Un message d'erreur ICMP DEVRAIT être envoyé avec les bits de TOS normaux (c'est-à-dire, zéro).

Un message d'erreur ICMP NE DOIT PAS être envoyé par suite de la réception de :

* un message d'erreur ICMP, ou
* un datagramme destiné à une adresse IP de diffusion ou de diffusion groupée, ou
* un datagramme envoyé comme diffusion de couche de liaison, ou
* un fragment non initial, ou
* un datagramme dont l'adresse de source ne définit pas un seul hôte -- par exemple, une adresse zéro, une adresse de bouclage arrière, une adresse de diffusion, une adresse de diffusion groupée, ou une adresse de classe E.

Note : Ces restrictions prennent le pas sur toute exigence figurant ailleurs dans le présent document pour l'envoi de messages d'erreur ICMP.

Discussion :
Ces règles empêcheront les "tempêtes de diffusion" qui ont résulté d'hôtes retournant les messages d'erreur ICMP en réponse aux datagrammes de diffusion. Par exemple, un segment UDP en diffusion à un accès non existant pourrait déclencher une inondation de datagrammes ICMP Destination injoignable de la part de toutes les machines qui n'ont pas de client pour cet accès de destination. Sur un grand Ethernet, les collisions résultantes peuvent rendre le réseau inutilisable pendant une seconde ou plus.

Tout datagramme qui est diffusé sur le réseau connecté devrait avoir une adresse de diffusion IP valide comme destination IP (voir au paragraphe 3.3.6). Cependant, certains hôtes violent cette règle. Pour être certain de détecter les datagrammes en diffusion, il est donc exigé que les hôtes vérifient s'il s'agit d'une adresse de diffusion de couche de liaison ou d'une adresse de diffusion de couche IP.

Mise en œuvre :
Cela exige que la couche de liaison informe la couche IP lorsque un datagramme de diffusion de couche de liaison a été reçu ; voir au paragraphe 2.4.

3.2.2.1 Destination injoignable : RFC-792

Les codes supplémentaires suivants sont définis ici :
6 = destination réseau inconnue
7 = hôte de destination inconnu
8 = hôte de source isolé

9 = communication administrativement interdite avec le réseau de destination
10 = communication administrativement interdite avec l'hôte de destination
11 = réseau injoignable pour ce type de service
12 = hôte injoignable pour ce type of service

Un hôte DEVRAIT générer les messages Destination injoignable avec le code :
2 (Protocole injoignable), lorsque le protocole de transport désigné n'est pas pris en charge ; ou
3 (Accès injoignable), lorsque le protocole de transport désigné (par exemple, UDP) est incapable de démultiplexer le datagramme mais n'a pas de mécanisme de protocole pour en informer l'envoyeur.

Un message Destination injoignable qui est reçu DOIT être rapporté à la couche transport. La couche transport DEVRAIT utiliser l'information de façon appropriée ; par exemple, voir aux paragraphes 4.1.3.3, 4.2.3.9, et 4.2.4 ci-dessous. Un protocole de transport qui a son propre mécanisme pour notifier à l'envoyeur qu'un accès est injoignable (par exemple, TCP, qui envoie des segments RST) DOIT néanmoins accepter un Accès injoignable ICMP à la place.

Un message Destination injoignable qui est reçu avec le code 0 (Réseau), 1 (Hôte), ou 5 (Mauvaise route de source) peut résulter d'un acheminement transitoire et DOIT donc seulement être interprété comme une indication, non une preuve, que la destination spécifiée est injoignable [RFC0816]. Par exemple, il NE DOIT PAS être utilisé comme preuve d'un routeur défaillant (voir au paragraphe 3.3.1).

3.2.2.2 Redirection : RFC-792

Un hôte NE DEVRAIT PAS envoyer un message ICMP Rediriger *(Redirect)* ; Rediriger ne doit être envoyé que par les routeurs.

Un hôte qui reçoit un message Rediriger DOIT mettre à jour ses informations d'acheminement en conséquence. Chaque hôte DOIT être prêt à accepter à la fois des Rediriger d'hôte et du réseau et à les traiter comme décrit au paragraphe 3.3.1.2 ci-dessous.

Un message Rediriger DEVRAIT être éliminé en silence si la nouvelle adresse de routeur qu'il spécifie n'est pas sur le même (sous-) réseau connecté sur lequel le Rediriger est arrivé ([RFC1009] Appendice A) ou si la source du Rediriger n'est pas le routeur de premier bond actuel pour la destination spécifiée (voir au paragraphe 3.3.1).

3.2.2.3 Source éteinte : RFC-792

Un hôte PEUT envoyer un message Source éteinte *(Source Quench)* si il approche, ou a atteint, le point auquel il est forcé d'éliminer des datagrammes entrants du fait qu'il est à court de mémoire tampon de réassemblage ou d'autres ressources. Voir au paragraphe 2.2.3 de la [RFC1009] des suggestions sur le moment où envoyer Source éteinte.

Si un message Source éteinte est reçu, la couche IP DOIT le rapporter à la couche transport (ou au traitement ICMP). En général, la couche transport ou application DEVRAIT mettre en œuvre un mécanisme pour répondre à Source éteinte pour tout protocole qui peut envoyer une séquence de datagrammes à la même destination dont on peut raisonnablement attendre qu'il conserve assez d'informations d'état pour le rendre faisable. Voir à la Section 4 le traitement de Source éteinte par TCP et UDP.

Discussion :
Un Source éteinte peut être généré par l'hôte cible ou par un routeur sur le chemin d'un datagramme. L'hôte qui reçoit un Source éteinte devrait se mettre au ralenti pendant un certain temps, puis à nouveau graduellement augmenter le débit de transmission. Le mécanisme de réponse à Source éteinte peut être dans la couche transport (pour les protocoles orientés connexion comme TCP) ou dans la couche d'application (pour les protocoles qui sont bâtis par dessus UDP).

Un mécanisme a été proposé [RFC1016] pour faire répondre directement la couche IP à Source éteinte en contrôlant le débit d'envoi des datagrammes, cependant, cette proposition est actuellement expérimentale et n'est pas recommandée.

3.2.2.4 Délai dépassé : RFC-792

Un message Délai dépassé *(Time Exceeded)* entrant DOIT être passé à la couche transport.

Discussion : Un routeur enverra un message Délai dépassé de code 0 (en transit) lorsque il élimine un datagramme du fait d'un champ TTL expiré. Cela indique soit une boucle d'acheminement de routeur soit une valeur initiale trop faible de TTL.

Un hôte peut recevoir un message Délai dépassé de code 1 (Temporisation de réassemblage expirée) de la part d'un hôte de destination qui a constaté la fin de temporisation d'un datagramme incomplet et l'a éliminé ; voir au paragraphe 3.3.2 ci-dessous. À l'avenir, la réception de ce message pourrait faire partie d'une sorte de procédure de "découverte de MTU", pour découvrir la taille maximum de datagramme qui peut être envoyée sans fragmentation sur le chemin.

3.2.2.5 Problème de paramètre : RFC-792

Un hôte DEVRAIT générer des messages Problème de paramètre *(Parameter Problem)*. Un message Problème de paramètre entrant DOIT être passé à la couche transport, et il PEUT être rapporté à l'utilisateur.

Discussion :
Le message ICMP Problème de paramètre est envoyé à l'hôte de source pour tout problème non spécifiquement couvert par un autre message ICMP. La réception d'un message Problème de paramètre indique généralement quelque erreur de mise en œuvre locale ou distante.

Une nouvelle variante du message Problème de paramètre est définie ici :
Code 1 = L'option requise manque.

Discussion :
Cette variante est actuellement utilisée dans la communauté militaire pour une option de sécurité manquante.

3.2.2.6 Demande/réponse d'écho : RFC-792

Tout hôte DOIT mettre en œuvre une fonction de serveur d'écho ICMP qui reçoive les demandes d'écho et envoie les réponses d'écho correspondantes. Un hôte DEVRAIT aussi mettre en œuvre une interface de couche d'application pour envoyer une demande d'écho et recevoir une réponse d'écho, pour les besoins de diagnostic.

Une demande d'écho ICMP destinée à une adresse de diffusion ou diffusion groupée IP PEUT être éliminée en silence.

Discussion :
Cette disposition neutre résulte d'un débat passionné entre ceux qui pensent qu'un écho ICMP à une adresse de diffusion fournit une capacité de diagnostic valable et ceux qui pensent qu'un détournement de l'utilisation de cette capacité peut facilement créer des tempêtes de paquets.

L'adresse de source IP dans une réponse d'écho ICMP DOIT être la même que l'adresse de destination spécifique (définie au paragraphe 3.2.1.3) du message de demande d'écho ICMP correspondant.

Les données reçues dans une demande d'écho ICMP DOIVENT être entièrement incluses dans la réponse d'écho résultante. Cependant, si l'envoi de la réponse d'écho exige une fragmentation intentionnelle qui n'est pas mise en œuvre, le datagramme DOIT être tronqué à la taille de transmission maximum (voir au paragraphe 3.3.3) et envoyé.

Les messages de réponse d'écho DOIVENT être passés à l'interface d'utilisateur ICMP, à moins que la demande d'écho correspondante ait été générée dans la couche IP.

Si une option Route enregistrée et/ou Horodatage est reçue dans une demande d'écho ICMP, cette ou ces options DEVRAIENT être mises à jour pour inclure l'hôte actuel, et incluses dans l'en-tête IP du message Réponse d'écho, sans "fractionnement". Et donc, le chemin enregistré sera pour l'aller-retour complet.

Si une option Route de source est reçue dans une demande d'écho ICMP, la route de retour DOIT être inversée et utilisée comme option Source pour le message de réponse d'écho.

3.2.2.7 Demande/réponse d'information : RFC-792

Un hôte NE DEVRAIT PAS mettre en œuvre ces messages.

Discussion : La paire Demande/Réponse d'information était destinée à prendre en charge les systèmes auto configurants tels que les stations de travail sans disque dur, pour leur permettre de découvrir leur numéro de réseau IP au moment de l'amorçage. Cependant, les protocoles RARP et BOOTP fournissent de meilleurs mécanismes pour qu'un hôte découvre sa propre adresse IP.

3.2.2.8 Horodatage et réponse d'horodatage : RFC-792

Un hôte PEUT mettre en œuvre Horodatage et Réponse d'horodatage. Si il les met en œuvre, les règles suivantes DOIVENT être suivies.

- La fonction de serveur Horodatage ICMP retourne une Réponse d'horodatage à chaque message Horodatage qui est reçu. Si cette fonction est mise en œuvre, elle DEVRAIT être conçue pour une variabilité minimale en délai (par exemple, mise en œuvre dans le noyau pour éviter des retards de la programmation d'un processus d'utilisateur).

Les cas suivants de Horodatage sont à traiter conformément aux règles correspondantes pour l'écho ICMP :

- Un message de demande d'horodatage ICMP à une adresse IP de diffusion ou diffusion groupée PEUT être éliminé en silence.

- L'adresse IP de source dans une réponse d'horodatage ICMP DOIT être la même que l'adresse de destination spécifique du message Demande d'horodatage correspondant.

- Si une option Route de source est reçue dans une demande d'écho ICMP, le chemin de retour DOIT être inversé et utilisé comme option de Route de source pour le message de réponse d'horodatage.

- Si une option Record Route et/ou Horodatage est reçue dans une demande d'horodatage, cette ou ces options DEVRAIENT être mises à jour pour inclure l'hôte actuel, et incluses dans l'en-tête IP du message Réponse d'horodatage.

- Les messages entrants de Réponse d'horodatage DOIVENT être passés à l'interface d'utilisateur ICMP.

La forme préférée pour une valeur d'horodatage (la "valeur standard") est en unités de millisecondes à partir de minuit en Temps Universel. Cependant, il peut être difficile de fournir cette valeur avec une résolution d'une milliseconde. Par exemple, de nombreux systèmes utilisent des horloges qui ne mettent à jour qu'à la fréquence de ligne, 50 ou 60 fois par seconde. Donc une certaine latitude est admise dans la "valeur standard":

(a) Une "valeur standard" DOIT être mise à jour au moins 15 fois par seconde (c'est-à-dire qu'au plus les six bits de moindre poids de la valeur peuvent être indéfinis).

(b) La précision d'une "valeur standard" DOIT approximer celle des horloges CPU réglées par l'opérateur, c'est-à-dire, correcte sur quelques minutes.

3.2.2.9 Demande/réponse de gabarit d'adresse : RFC-950

Un hôte DOIT prendre en charge la première, et PEUT mettre en œuvre les trois méthodes suivantes pour déterminer le ou les gabarits d'adresse correspondants à sa ou ses adresses IP :

(1) informations de configuration statique ;
(2) obtention dynamique du ou des gabarits d'adresse en sous-produit du processus d'initialisation du système (voir la [RFC1123]) ;
(3) envoi de demandes de gabarit d'adresse ICMP et réception de réponses de gabarit d'adresse ICMP.

Le choix de la méthode à utiliser dans un hôte particulier DOIT être configurable.

Lorsque la méthode (3) (utilisation des messages Gabarit d'adresse) est activée, alors :

(a) À l'initialisation, l'hôte DOIT diffuser un message Demande de gabarit d'adresse sur le réseau connecté correspondant à l'adresse IP. Il DOIT retransmettre ce message un petit nombre de fois s'il ne reçoit pas une Réponse de gabarit d'adresse immédiate.
(b) Jusqu'à ce qu'il ait reçu une Réponse de gabarit d'adresse, l'hôte DEVRAIT supposer un gabarit approprié pour la classe d'adresses de l'adresse IP, c'est-à-dire, supposer que le réseau connecté n'est pas subdivisé en sous-réseaux.

(c) Le premier message de Réponse de gabarit d'adresse reçu DOIT être utilisé pour établir le gabarit d'adresse correspondant à l'adresse IP locale particulière. Ceci est vrai même si le premier message Réponse de gabarit d'adresse est "non sollicité", auquel cas il aura été diffusé et peut arriver après que l'hôte ait cessé de réémettre des Demandes de gabarit d'adresse. Une fois que le gabarit a été établi par une Réponse de gabarit d'adresse, les messages ultérieurs de Réponse de gabarit d'adresse DOIVENT être ignorés (en silence).

À l'inverse, si les messages de gabarit d'adresse sont désactivés, aucune Demande de gabarit d'adresse ICMP ne sera envoyée, et toute Réponse de gabarit d'adresse ICMP reçue pour cette adresse IP locale DOIT être ignorée (en silence).

Un hôte DEVRAIT faire des vérifications de vraisemblance sur tout gabarit d'adresse qu'il installe ; voir au paragraphe Mise en œuvre ci-dessous.

Un système NE DOIT PAS envoyer de Réponse de gabarit d'adresse s'il n'est pas un agent d'autorisation pour les gabarits d'adresse. Un agent d'autorisation peut être un hôte ou un routeur, mais il DOIT être explicitement configuré comme agent de gabarit d'adresse. La réception d'un gabarit d'adresse via une Réponse de gabarit d'adresse ne donne pas autorité au receveur et NE DOIT PAS être utilisée comme fondement de la production de Réponses de gabarit d'adresse.

Avec une configuration statique de gabarit d'adresse, il DEVRAIT y avoir un fanion de configuration supplémentaire qui détermine si l'hôte va agir comme agent d'autorisation pour ce gabarit, c'est-à-dire, si il va répondre à ces messages Demande de gabarit d'adresse en utilisant ce gabarit.

Si il est configuré comme agent, l'hôte DOIT diffuser une Réponse de gabarit d'adresse pour le gabarit sur l'interface appropriée lors de l'initialisation.

Voir "Initialisation du système" dans la [RFC1123] pour des informations complémentaires sur l'utilisation des messages de Demande/Réponse de gabarit d'adresse.

Discussion :
Les hôtes qui envoient au hasard des réponses de gabarit d'adresse avec des gabarits d'adresse invalides ont souvent constitué une nuisance sérieuse. Pour empêcher cela, les réponses de gabarit d'adresse ne devraient être envoyées que par des agents d'autorisation qui ont été choisis par une action administrative explicite.

Lorsque un agent d'autorisation reçoit un message Demande de gabarit d'adresse, il va envoyer une Réponse de gabarit d'adresse en envoi individuel à l'adresse IP de source. Si la partie réseau de cette adresse est zéro (voir (a) et (b) au paragraphe 3.2.1.3), la réponse sera en diffusion.

S'il ne reçoit pas de réponse à ses messages Demande de gabarit d'adresse, un hôte va supposer qu'il n'y a pas d'agent et utilisera un gabarit sans sous-réseau, mais l'agent peut n'avoir été que temporairement injoignable. Un agent diffusera une Réponse de gabarit d'adresse non sollicitée chaque fois qu'il s'initialise, afin de mettre à jour les gabarits de tous les hôtes qui se sont initialisés dans l'intervalle.

Mise en œuvre :
La vérification de vraisemblance suivante sur un gabarit d'adresse est suggérée : le gabarit n'est pas de bits tous à un, et il est soit de zéro soit alors les huit bits de plus fort poids sont à un.

3.2.3 Protocole de gestion de groupe Internet (IGMP)

IGMP [RFC1112] est un protocole utilisé entre hôtes et routeurs sur un seul réseau pour établir la liste de membre des hôtes dans les groupes de diffusion groupée particuliers. Les routeurs utilisent ces informations, en conjonction avec un protocole d'acheminement de diffusion groupée, pour prendre en charge la diffusion groupée IP à travers l'Internet.

Pour le moment, la mise en œuvre de IGMP est FACULTATIVE ; voir au paragraphe 3.3.7 pour des informations complémentaires. Sans IGMP, un hôte peut toujours participer à des diffusions groupées locales à ses réseaux connectés.

3.3 Questions spécifiques

3.3.1 Datagrammes d'acheminement sortant

La couche IP choisit le prochain bond correct pour chaque datagramme qu'elle envoie. Si la destination est sur un réseau connecté, le datagramme est envoyé directement à l'hôte de destination ; autrement, il doit être acheminé à un routeur sur un réseau connecté.

3.3.1.1 Décision locale/distante

Pour décider si la destination est sur un réseau connecté, l'algorithme suivant DOIT être utilisé (voir la [RFC0950]) :

(a) Le gabarit d'adresse (particulier d'une adresse IP locale pour un hôte à multi-rattachement) est un gabarit de 32 bits qui choisit les champs Numéro de réseau et Numéro de sous-réseau de l'adresse IP correspondante.

(b) Si les bits de l'adresse de destination IP extraits par le gabarit d'adresse correspondent aux bits de l'adresse de source IP extraits par le même gabarit, la destination est alors sur le réseau connecté correspondant, et le datagramme est à transmettre directement à l'hôte de destination.

(c) Sinon, la destination n'est alors accessible qu'à travers une passerelle. Le choix d'une passerelle est décrit ci-dessous (3.3.1.2).

Un cas particulier d'adresse de destination est traité comme suit :
* Pour une adresse de diffusion ou diffusion groupée limitée, passer simplement le datagramme à la couche de liaison vers l'interface appropriée.
* Pour une diffusion dirigée (réseau ou sous-réseau), le datagramme peut utiliser les algorithmes d'acheminement standard.

La couche IP d'hôte DOIT fonctionner correctement dans un environnement réseau minimal, et en particulier, lorsque il n'y a pas de routeur. Par exemple, si la couche IP d'un hôte insiste pour trouver au moins un routeur à initialiser, l'hôte sera incapable de fonctionner sur un seul réseau de diffusion isolé.

3.3.1.2 Choix de passerelle (de routeur)

Pour acheminer efficacement une série de datagrammes à la même destination, l'hôte de source DOIT conserver une "antémémoire des routes" de la cartographie des routeurs de prochain bond. Un hôte utilise l'algorithme de base suivant sur son antémémoire pour acheminer un datagramme ; cet algorithme est conçu pour faire porter le principal fardeau de l'acheminement sur les routeurs [RFC0816].

(a) Si l'antémémoire des routes ne contient pas d'information pour une destination particulière, l'hôte choisit un routeur "par défaut" et lui envoie le datagramme. Elle construit aussi une entrée d'antémémoire des routes correspondante.

(b) Si ce routeur n'est pas le meilleur prochain bond pour la destination, le routeur va transmettre le datagramme au routeur du meilleur prochain bond et retourner un message Rediriger ICMP à l'hôte de source.

(c) Lorsqu'il reçoit un Rediriger, l'hôte met à jour le routeur du prochain bond dans l'entrée appropriée de l'antémémoire des routes, de sorte que les datagrammes ultérieurs pour la même destination aillent directement au meilleur routeur.

Comme le gabarit de sous-réseau approprié à l'adresse de destination n'est généralement pas connu, un message Redirection réseau *(Network Redirect)* DEVRAIT être traité de façon identique à celle d'un message Redirection d'hôte *(Host Redirect)* ; c'est-à-dire que (seule) l'entrée d'antémémoire pour l'hôte de destination serait mise à jour (ou créée s'il n'existe pas d'entrée pour cet hôte) pour le nouveau routeur.

Discussion :
Cette recommandation est destinée à protéger contre les routeurs qui envoient par erreur des Redirection réseau pour un réseau en sous-réseaux, en violation des exigences pour les routeurs [RFC1009].

Lorsque il n'y a pas d'entrée d'antémémoire des routes pour l'adresse de l'hôte de destination (et que la destination n'est pas sur le réseau connecté), la couche IP DOIT prendre un routeur sur sa liste de routeurs "par défaut". La couche IP DOIT prendre en charge plusieurs routeurs par défaut.

Comme caractéristique supplémentaire, une couche IP d'hôte PEUT mettre en œuvre un tableau des "routes statiques". Chacune de ces routes statiques PEUT inclure un fanion qui spécifie si elle peut être subrogée par des Rediriger ICMP.

Discussion :
Un hôte a généralement besoin de connaître au moins un routeur par défaut pour démarrer. Cette information peut être obtenue d'un fichier de configuration ou bien d'une séquence de démarrage d'hôte, par exemple, le protocole BOOTP (voir la [RFC1123]).

Il a été suggéré qu'un hôte puisse augmenter sa liste de routeurs par défaut en enregistrant tout nouveau routeur dont il apprend l'existence. Par exemple, il peut enregistrer tout routeur sur lequel il est redirigé. Une telle caractéristique, bien que pouvant être utile dans certaines circonstances, peut causer des problèmes dans d'autres cas (par exemple, les routeurs ne sont pas toujours égaux) et n'est pas recommandée.

Une route statique est normalement une transposition particulière pré-établie entre un hôte ou réseau de destination et un routeur particulier de prochain bond ; elle pourrait aussi dépendre du Type-de-Service (voir la section suivante). Les routes statiques sont établies par les administrateurs de système pour supplanter le mécanisme normal d'acheminement automatique, afin de traiter des situations exceptionnelles. Cependant, toute information d'acheminement statique est une source potentielle d'échec lors d'un changement de configuration ou d'une défaillance d'un équipement.

3.3.1.3 Antémémoire des routes

Chaque entrée d'antémémoire des routes doit inclure les champs suivants :
(1) Adresse IP locale (pour un hôte multi rattachement)
(2) Adresse IP de destination
(3) Type(s)-de-Service (TOS)
(4) Adresse IP du routeur du prochain bond

Le champ (2) PEUT être l'adresse IP complète de l'hôte de destination, ou seulement le numéro du réseau de destination. Le champ (3), TOS, DEVRAIT être inclus.

Voir au paragraphe 3.3.4.2 l'exposé sur les implications du multi rattachement sur la procédure de recherche dans cette antémémoire.

Discussion :
L'inclusion du champ Type-de-Service dans l'antémémoire des routes et sa prise en compte dans l'algorithme d'acheminement d'hôte fournira les mécanismes nécessaires à l'avenir lorsque l'acheminement par Type-de-Service sera d'utilisation courante dans l'Internet. Voir au paragraphe 3.2.1.6.

Chaque entrée d'antémémoire des routes définit les points d'extrémité d'un chemin Internet. Bien que les chemins de connexion puissent changer arbitrairement de façon dynamique, les caractéristiques de transmission du chemin tendent à rester approximativement constantes sur une période plus longue que celle de la durée d'une seule connexion de transport d'hôte à hôte normale. Donc, une entrée d'antémémoire des routes est un endroit naturel pour mémoriser des données sur les propriétés du chemin. Des exemples de telles propriétés pourraient être la taille maximum de datagramme non fragmenté (voir au paragraphe 3.3.3) ou le délai moyen d'aller-retour mesuré par un protocole de transport. Ces données seront généralement à la fois rassemblées et utilisées par un protocole de couche supérieure, par exemple, par TCP, ou par une application utilisant UDP. Des expérimentations sont actuellement en cours sur cette façon de mettre en antémémoire les propriétés du chemin.

Il n'y a pas de consensus sur le fait de savoir si l'antémémoire des routes devrait être entrée seulement sur les adresses d'hôte de destination ou si l'on doit admettre à la fois les adresses d'hôte et de réseau. Ceux qui sont en faveur de l'utilisation des seules adresses d'hôte vont valoir que :
(1) Comme exigé au paragraphe 3.3.1.2, les messages Redirect vont généralement résulter en entrées frappées sur les adresses d'hôte de destination ; le schéma le plus simple et le plus général serait d'utiliser toujours les adresses d'hôte.
(2) La couche IP peut ne pas toujours connaître le gabarit d'adresse d'une adresse réseau dans un environnement complexe de sous-réseaux.
(3) L'utilisation des seules adresses d'hôte permet que l'adresse de destination soit utilisée comme pur numéro de 32 bits, ce qui pourrait permettre à l'avenir d'étendre plus facilement l'architecture de l'Internet sans aucun changement aux hôtes.

Le point de vue opposé est qu'admettre un mélange d'hôtes et de réseaux de destination dans l'antémémoire des routes :
(1) Économise de l'espace mémoire.
(2) Conduit à une structure des données plus simple, combinant facilement l'antémémoire avec les tableaux des routes par défaut et des routes statiques (voir ci-dessous).
(3) Fournit un endroit plus utile pour placer en antémémoire les propriétés des chemins, comme exposé plus haut.

Mise en œuvre :
L'antémémoire doit être assez vaste pour inclure les entrées pour le nombre maximum d'hôtes de destination qui peuvent être utilisés à la fois.

Une entrée d'antémémoire des routes peut aussi comporter des informations de commandes utilisées pour choisir une entrée de remplacement. Cela peut prendre la forme d'un bit "utilisation récente", d'un compte d'utilisation, ou d'un horodatage de dernière utilisation, par exemple. Il est recommandé que cela inclue l'heure de la dernière modification de l'entrée, pour les besoins de diagnostic.

Une mise en œuvre peut souhaiter réduire la redondance de l'examen de l'antémémoire des routes pour chaque datagramme à transmettre. Cela peut être accompli avec un tableau de hachage pour accélérer la recherche, ou en donnant à un protocole de transport orienté connexion un "conseil" ou un raccourci temporaire vers l'entrée appropriée d'antémémoire, à passer à la couche IP avec chaque datagramme suivant.

Bien que nous ayons décrit l'antémémoire des routes, la liste des routeurs par défaut, et un tableau des routes statiques comme des concepts distincts, ils peuvent en pratique être combinés dans une seule structure de données de "tableau d'acheminement".

3.3.1.4 Détection d'une défaillance de passerelle

La couche IP DOIT être capable de détecter la défaillance du routeur du "prochain bond" qui figure sur sa liste dans l'antémémoire et de choisir un routeur de remplacement (voir au paragraphe 3.3.1.5).

La détection des routeurs morts est traitée dans la [RFC0816] avec un certain détail. L'expérience jusqu'à ce jour n'a pas produit un algorithme complet qui soit totalement satisfaisant, bien qu'il ait identifié plusieurs chemins interdits et des techniques prometteuses.

* Un routeur particulier NE DEVRAIT PAS être utilisé indéfiniment en l'absence d'indications positives de son fonctionnement.
* Des preuves actives telles que le "ping" (c'est-à-dire, l'utilisation d'un échange ICMP de demande/réponse d'écho) sont coûteuses et peu significatives. En particulier, les hôtes NE DOIVENT PAS vérifier activement l'état d'un routeur de prochain bond simplement en "pingant" continuellement le routeur.
* Même lorsque c'est le seul moyen efficace pour vérifier l'état d'un routeur, le ping ne DOIT être utilisé que lorsque du trafic est envoyé au routeur et lorsque il n'y a pas d'autre indication positive pour suggérer que le routeur fonctionne.
* Pour éviter le ping, les couches au-dessus et/ou au-dessous de la couche Internet DEVRAIENT être capables de donner un "avis" sur l'état des entrées d'antémémoire des routes lorsque des informations positives (routeur OK) ou négatives (routeur mort) sont disponibles.

Discussion :
Si une mise en œuvre ne comporte pas un mécanisme adéquat pour détecter un routeur mort et un réacheminement, la défaillance d'un routeur peut causer la disparition apparente de datagrammes dans un "trou noir". Cette défaillance peut être extrêmement perturbante pour les utilisateurs et difficile à corriger pour le personnel du réseau.

Le mécanisme de détection de routeur mort ne doit pas causer une charge inacceptable pour l'hôte, pour les réseaux connectés, ou sur le ou les routeurs du prochain bond. Les contraintes exactes sur la détection à temps de routeurs morts et sur la charge acceptable peuvent varier un peu selon la nature de la mission de l'hôte, mais un hôte a généralement besoin de détecter assez rapidement la défaillance d'un routeur de prochain bond pour que les connexions de la couche transport ne se coupent pas avant qu'on puisse choisir un routeur de remplacement.

Passer des avis des autres couches de la pile de protocole complique les interfaces entre les couches, mais c'est l'approche préférée pour la détection de routeurs morts. L'avis peut provenir de presque toutes les parties de l'architecture TCP/IP, mais on s'attend à ce qu'il provienne principalement des couches transport et liaison. Voici quelques sources possibles pour les avis de routeurs morts :

- o TCP ou tout protocole de transport orienté connexion devrait être capable de donner un avis négatif, par exemple, déclenché par des retransmissions excessives.
- o TCP peut donner un avis positif lorsque des (nouvelles) données sont acquittées. Même si le chemin est asymétrique, un ACK pour les nouvelles données prouve que les données dont il est accusé réception doivent avoir été transmises avec succès.
- o Un message ICMP Redirect pour un routeur particulier devrait être utilisé comme avis positif sur ce routeur.
- o Les informations de couche liaison qui détectent et rapportent de façon fiable des défaillances d'hôte (par exemple, les messages ARPANET Destination Dead) devraient être utilisés comme avis négatif.
- o L'échec de transposition ARP ou de revalidation ARP peut être utilisé comme avis négatif pour l'adresse IP correspondante.
- o Les paquets qui arrivent d'une adresse de couche de liaison particulière sont la preuve que le système à cette adresse est vivant. Cependant, transformer cette information en un avis sur les routeurs exige de transposer l'adresse de couche de liaison en une adresse IP, puis de vérifier cette adresse IP par rapport aux routeurs pointés par l'antémémoire de routes. Ceci est probablement prohibitivement inefficace.

Noter qu'un avis positif donné pour chaque datagramme reçu peut causer une redondance inacceptable dans la mise en œuvre.

Bien que l'avis puisse être passé en utilisant les arguments requis dans toutes les interfaces à la couche IP, certains protocoles de couche transport et application ne peuvent pas déduire l'avis correct. Ces interfaces doivent donc permettre une valeur neutre comme avis, car un avis toujours positif ou toujours négatif conduit à des comportements incorrects.

Il y a une autre technique pour la détection de routeur mort qui a été couramment utilisée mais n'est pas recommandée.
Cette technique s'appuie sur la réception passive ("écoute") par l'hôte des datagrammes du protocole de passerelle intérieure (IGP, *Interior Gateway Protocol*) que les routeurs se diffusent les uns aux autres. Cette approche a l'inconvénient qu'un hôte a besoin de reconnaître tous les protocoles de passerelle intérieure que les routeurs peuvent utiliser (voir [RFC1009]). De plus, cela ne fonctionne que sur un réseau de diffusion.

À présent, le ping (c'est-à-dire, l'utilisation des messages d'écho ICMP) est le mécanisme de vérification des routeurs lorsqu'il est absolument nécessaire. Un ping réussi garantit que l'interface visée et sa machine associée sont actives, mais il ne garantit pas que la machine est un routeur plutôt qu'un hôte. La déduction normale est que si un Redirect ou autre évidence indique qu'une machine est un routeur, des ping réussis vont indiquer que la machine est toujours active et donc que c'est un routeur. Cependant, comme un hôte élimine en silence des paquets qu'un routeur transmettrait ou redirigerait, cette hypothèse peut parfois s'avérer fausse. Pour éviter ce problème, un nouveau message ICMP en cours de développement demandera "êtes vous un routeur ?"

Mise en œuvre :
L'algorithme spécifique suivant a été suggéré :

- o Associer un "temporisateur de réacheminement" à chaque routeur pointé sur l'antémémoire des routes. Initialiser le temporisateur à une valeur Tr, qui doit être assez petite pour permettre la détection d'un routeur mort avant la fin de temporisation des connexions de transport.
- o Un avis positif remet le temporisateur de réacheminement à Tr. Un avis négatif réduit le temporisateur de réacheminement ou le ramène à zéro.
- o Chaque fois que la couche IP utilise un routeur particulier pour acheminer un datagramme, elle va vérifier le temporisateur de réacheminement correspondant. Si le temporisateur est arrivé à expiration (a atteint zéro) la couche IP va envoyer un ping au routeur, suivi immédiatement par le datagramme.
- o Le ping (écho ICMP) sera renvoyé si nécessaire, jusqu'à N fois. Si aucune réponse ping n'est reçue en N essais, le routeur sera supposé défaillant, et un nouveau routeur de prochain bond sera choisi pour toutes les entrées de mémoire cache qui pointent sur le routeur défaillant.

Noter que la taille de Tr est inversement proportionnelle à la quantité d'avis disponibles. Tr devrait être assez grand pour garantir que :

* Tout envoi de ping restera à un bas niveau (par exemple, < 10 %) de tous les paquets envoyés à un routeur de la part de l'hôte, ET
* L'utilisation du ping est peu fréquente (par exemple, toutes les 3 minutes)

Comme l'algorithme recommandé est en rapport avec les routeurs pointés par les entrées d'antémémoire des routes, plutôt qu'avec les entrées d'antémémoire elles-mêmes, une structure de données à deux niveaux (peut-être coordonnés par des antémémoires ARP ou similaires) peut être souhaitable pour la mise en œuvre d'une antémémoire des routes.

3.3.1.5 Choix d'un nouveau routeur

Si le routeur défaillant n'est pas celui actuellement par défaut, la couche IP peut immédiatement passer sur un routeur par défaut. Si c'est le routeur par défaut qui est défaillant, la couche IP DOIT choisir un routeur par défaut différent (en supposant qu'elle connaît plus d'un routeur par défaut) pour la route défaillante et pour établir de nouvelles routes.

Discussion :
Lors de la défaillance d'un routeur, les autres routeurs sur le réseau connecté vont connaître la défaillance par le biais d'un protocole d'acheminement inter-routeurs. Cependant, cela ne va pas se faire instantanément, car les protocoles d'acheminement de routeurs ont normalement un temps d'établissement de 30 à 60 secondes. Si l'hôte passe à un routeur de remplacement avant que les routeurs se soient mis d'accord sur la défaillance, le nouveau routeur cible va probablement transmettre le datagramme au routeur défaillant en renvoyer un Redirect à l'hôte pointant sur le routeur défaillant (!). Le résultat sera vraisemblablement une oscillation rapide des contenus de l'antémémoire des routes de l'hôte pendant la période d'établissement du routeur. Il avait été proposé que le logiciel du routeur mort devrait comporter un mécanisme d'hystérésis pour empêcher de telles oscillations. Cependant, l'expérience n'a pas démontré de dommages résultant de telles oscillations, car le service ne peut être restauré avec l'hôte jusqu'à ce que les informations d'acheminement de l'hôte se soient stabilisées.

Mise en œuvre :
Une technique de mise en œuvre pour le choix d'un nouveau routeur par défaut est de faire un simple round-robin parmi les routeurs par défaut de la liste de l'hôte. Une autre est de ranger les routeurs par ordre de priorité, et quand le routeur par défaut actuel n'est plus celui du plus fort rang de priorité, de faire lentement un "ping" des routeurs de priorité élevée pour détecter quand revenir en service. Cet envoi de ping peut être à très faible débit, par exemple, de 0,005 par seconde.

3.3.1.6 Initialisation

Les informations suivantes DOIVENT être configurables :
(1) la ou les adresses IP.
(2) le ou les gabarits d'adresse.
(3) une liste des routeurs par défaut, avec un niveau de préférence.

Une méthode manuelle d'entrée de ces données de configuration DOIT être fournie. De plus, diverses méthodes peuvent être utilisées pour déterminer ces informations de façon dynamique; voir le paragraphe sur "Initialisation d'hôte" dans la [RFC1123].

Discussion :
Certaines mises en œuvre d'hôte utilisent "l'écoute" des protocoles de routeur sur un réseau de diffusion pour apprendre quels routeurs existent. Une méthode standard pour la découverte du routeur par défaut est en cours de développement.

3.3.2 Réassemblage

La couche IP DOIT mettre en œuvre le réassemblage des datagrammes IP.

On désigne la plus grande taille de datagramme qui peut être réassemblée par EMTU_R ("MTU effective à recevoir") ; elle est parfois appelée la "taille de réassemblage de mémoire tampon". L'EMTU_R DOIT être supérieure ou égale à 576, et DEVRAIT être configurable ou indéfinie, et DEVRAIT être supérieure ou égale à la MTU du ou des réseaux connectés.

Discussion :
Une limite fixe d'EMTU_R ne devrait pas être incorporée dans le code parce que certains protocoles de couche d'application exigent des valeurs de EMTU_R supérieures à 576.

Mise en œuvre :
Une mise en œuvre peut utiliser une mémoire tampon de réassemblage contiguë pour chaque datagramme, ou elle peut utiliser une structure de données plus complexe qui ne fixe pas de limite définie à la taille du datagramme réassemblé ; dans ce dernier cas, EMTU_R est dit être "indéfinie".

Logiquement, le réassemblage est effectué par la simple copie de chaque fragment dans la mémoire tampon de paquet au décalage approprié. Noter que les fragments peuvent se chevaucher si des retransmissions successives utilisent une mise en paquet différente et le même identifiant de réassemblage.

La partie difficile du réassemblage est la comptabilité pour déterminer quand tous les octets du datagramme ont été réassemblés. On recommande l'algorithme de Clark [RFC0815] qui n'exige pas d'espace de données additionnel pour la comptabilité. Cependant, noter que, contrairement à la [RFC0815], le premier en-tête de fragment doit être sauvegardé pour inclusion dans un possible message ICMP de dépassement de temps (Expiration de délai de réassemblage).

Il DOIT y avoir un mécanisme par lequel la couche transport puisse apprendre MMS_R, la taille maximum de message qui peut être reçue et réassemblée dans un datagramme IP (voir les appels GET_MAXSIZES au paragraphe 3.4). Si EMTU_R n'est pas indéfinie, la valeur de MMS_R est alors donnée par :

MMS_R = EMTU_R - 20

car 20 est la taille minimum d'un en-tête IP.

Il DOIT y avoir une temporisation de réassemblage. La valeur de la temporisation de réassemblage DEVRAIT être une valeur fixée, non établie par le TTL restant. Il est recommandé que cette valeur se tienne entre 60 et 120 secondes. Si la temporisation arrive à expiration, le datagramme partiellement réassemblé DOIT être éliminé et un message ICMP Délai dépassé être envoyé à l'hôte de source (si le fragment zéro a été reçu).

Discussion :
La spécification IP dit que la temporisation de réassemblage devrait être le TTL restant d'après l'en-tête IP, mais cela ne fonctionne pas bien parce que les routeurs traitent généralement le TTL comme un simple compte de bonds plutôt qu'un temps écoulé. Si la temporisation de réassemblage est trop courte, les datagrammes seront éliminés sans nécessité, et la communication peut échouer. La temporisation doit être au moins aussi longue que le délai maximum normal à travers l'Internet. Un minimum réaliste de temporisation de réassemblage serait de 60 secondes.

Il a été suggéré qu'il pourrait être tenu une antémémoire des temps d'aller-retour mesurés par les protocoles de transport pour les diverses destinations, et que ces valeurs pourraient être utilisées pour déterminer de façon dynamique une valeur raisonnable de temporisation de réassemblage. Une investigation plus approfondie de cette approche est nécessaire.

Si la temporisation de réassemblage est trop longue, les ressources de mémoire tampon chez l'hôte de réception seront liées pendant trop longtemps, et le MSL (*Maximum Segment Lifetime*, durée de vie maximum de segment) [RFC0793] seront plus grandes que nécessaire. Le MSL contrôle le débit maximum auquel les datagrammes fragmentés peuvent être envoyés en utilisant des valeurs distinctes du champ Identifiant de 16 bits ; une plus grande MSL diminue le débit maximum. La spécification TCP [RFC0793] suppose arbitrairement une valeur de 2 minutes pour MSL. Cela établit une limite supérieure à une valeur raisonnable de temporisation de réassemblage.

3.3.3 Fragmentation

Facultativement, la couche IP PEUT mettre en œuvre un mécanisme pour fragmenter intentionnellement les datagrammes sortants.

On désigne par EMTU_S (*"Effective MTU for sending"*, MTU effective d'envoi) la taille maximum de datagramme IP qui peut être envoyée, pour une combinaison particulière d'adresses IP de source et de destination et peut-être de TOS.

Un hôte DOIT mettre en œuvre un mécanisme permettant à la couche transport d'apprendre MMS_S, la taille maximum de message de couche transport qu'il peut envoyer pour un triplet donné de {source, destination, TOS} (voir l'appel GET_MAXSIZES au paragraphe 3.4). Si aucune fragmentation locale n'est effectuée, la valeur de MMS_S sera :

MMS_S = EMTU_S - <taille d'en-tête IP>

et EMTU_S doit être inférieur ou égal à la MTU de l'interface réseau correspondant à l'adresse de source du datagramme.

Noter que <taille d'en-tête IP> dans cette équation sera 20, sauf si le protocole IP réserve de l'espace pour insérer des options IP à ses propres fins en plus de toutes les options insérées par la couche transport.

Un hôte qui ne met pas en œuvre la fragmentation locale DOIT s'assurer que la couche transport (pour TCP) ou la couche d'application (pour UDP) obtient MMS_S de la couche IP et n'envoie pas un datagramme excédant MMS_S en taille.

Il est généralement souhaitable d'éviter la fragmentation locale et de choisir EMTU_S assez basse pour éviter la fragmentation dans les routeurs le long du chemin. En l'absence de connaissance réelle de la MTU minimum le long du

chemin, la couche IP DEVRAIT utiliser EMTU_S ≤ 576 chaque fois que l'adresse de destination n'est pas sur un réseau connecté, et autrement utiliser la MTU du réseau connecté.

La MTU de chaque interface physique DOIT être configurable.

Une mise en œuvre de couche IP d'hôte PEUT avoir un fanion de configuration "MTU-tous-sous-réseaux", indiquant que la MTU du réseau connecté est à utiliser pour les destinations sur les différents sous-réseaux au sein du même réseau, mais pas pour les autres réseaux. Et donc, ce fanion cause l'utilisation du gabarit de classe de réseau, plutôt que le gabarit d'adresse de sous-réseau, pour choisir un EMTU_S. Pour un hôte à rattachements multiples, un fanion "MTU-tous-sous-réseaux" est nécessaire pour chaque interface réseau.

Discussion :
Saisir la taille correcte de datagramme à utiliser pour l'envoi de données est un sujet complexe [IP:9].

(a) En général, aucun hôte n'est obligé d'accepter un datagramme IP plus long que 576 octets (incluant l'en-tête et les données), aussi un hôte ne doit pas envoyer de plus grand datagramme sans connaissance explicite ou sans accord préalable avec l'hôte de destination. Et donc, MMS_S est seulement la limite supérieure de la taille de datagramme qu'un protocole de transport peut envoyer ; même lorsque MMS_S dépasse 556, la couche transport doit limiter ses messages à 556 octets en l'absence d'autres informations sur l'hôte de destination.

(b) Certains protocoles de transport (par exemple, TCP) donnent le moyen d'informer explicitement l'envoyeur sur le plus grand datagramme que l'autre extrémité peut recevoir et réassembler [RFC0879]. Il n'y a pas de mécanisme correspondant à la couche IP.
Un protocole de transport qui suppose une EMTU_R supérieure à 576 (voir au paragraphe 3.3.2), peut envoyer un datagramme de cette taille supérieure à un autre hôte qui met en œuvre le même protocole.

(c) Dans l'idéal, les hôtes devrait limiter leur EMTU_S pour une destination donnée à la MTU minimum de tous les réseaux le long du chemin, pour éviter toute fragmentation. La fragmentation IP, bien que formellement correcte, peut créer de sérieux problèmes de performances de protocole de transport, parce que la perte d'un seul fragment signifie que tous les fragments du segment doivent être retransmis [IP:9].

Comme presque tous les réseaux de l'Internet acceptent actuellement une MTU de 576 ou plus, nous recommandons vivement l'utilisation de 576 pour les datagrammes envoyés à des réseaux non locaux.

Il a été suggéré qu'un hôte pourrait déterminer la MTU sur un chemin donné en envoyant un fragment de datagramme de décalage zéro et en attendant que le receveur excède le délai de réassemblage (qu'il ne peut achever !) et retourne un message ICMP Délai dépassé. Ce message inclurait le plus grand en-tête de fragment restant dans son corps. Des mécanismes plus directs sont en cours d'expérimentation, mais ne sont pas encore adoptés (voir par exemple, la RFC-1063).

3.3.4 Multi rattachement local

3.3.4.1 Introduction

Un hôte a rattachements multiples a plusieurs adresses IP, qu'on peut voir comme des "interfaces logiques". Ces interfaces logiques peuvent être associées à une ou plusieurs interfaces physiques, et ces interfaces physiques peuvent être connectées au même réseau ou à des réseaux différents.

Voici quelques cas importants de rattachement multiple :

(a) Plusieurs réseaux logiques
Les architectes de l'Internet supposaient que chaque réseau physique aurait un seul et unique numéro de réseau (ou sous-réseau) IP. Cependant, les administrateurs de LAN ont parfois trouvé utile de violer cette hypothèse, en faisant fonctionner un LAN avec plusieurs réseaux logiques sur un réseau connecté physique.

Si un hôte connecté à un tel réseau physique est configuré de façon à traiter du trafic pour chacun des N différents réseaux logiques, l'hôte aura alors N interfaces logiques. Celles-ci pourraient partager une seule interface physique, ou utiliser N interfaces physiques avec le même réseau.

(b) Plusieurs hôtes logiques

Lorsque un hôte a plusieurs adresses IP qui ont toutes la même partie <Numéro-de-réseau> (et la même partie <Numéro-de-sous-réseau>, s'il en est) les interfaces logiques sont appelées "hôtes logiques". Ces interfaces logiques pourraient partager une seule interface physique ou pourraient utiliser des interfaces physiques distinctes avec le même réseau physique.

(c) Multi rattachement simple
Dans ce cas, chaque interface logique est transposée en une interface physique séparée et chaque interface physique est connectée à un réseau physique différent. Le terme "multi rattachement" n'était à l'origine appliqué qu'à ce cas, mais il a maintenant un sens plus général.

Un hôte ayant une fonction de routeur incorporée va normalement entrer dans le cas du multi rattachement simple. Noter, cependant, qu'un hôte peut être à multi rattachement simple sans contenir de routeur incorporé, c'est-à-dire, sans transmettre de datagrammes d'un réseau connecté à un autre.

Ce cas présente les problèmes d'acheminement les plus difficiles. Le choix de l'interface (c'est-à-dire, le choix du réseau du premier bond) peut significativement affecter les performances ou même l'accessibilité de parties éloignées de l'Internet.

Finalement, on note une autre possibilité qui N'EST PAS du multi rattachement : une interface logique peut être liée à plusieurs interfaces physiques, afin d'accroître la fiabilité ou le débit entre des machines directement connectées en fournissant des chemins physiques de remplacement entre elles. Par exemple, deux systèmes pourraient être connectés par plusieurs liaisons point à point. On appelle cela "multiplexage de couche de liaison". Avec le multiplexage de couche de liaison, les protocoles au-dessus de la couche de liaison ne savent pas que plusieurs interfaces physiques sont présentes ; le pilote d'appareil de couche de liaison est chargé de multiplexer et router les paquets à travers les interfaces physiques.

Dans l'architecture du protocole Internet, une instance de protocole de transport ("entité") n'a pas d'adresse en propre, mais utilise à la place une seule adresse de protocole Internet (IP). Ceci a des implications pour les couches IP, transport, et d'application, et pour les interfaces entre elles. En particulier, le logiciel d'application peut avoir besoin de connaître les multiples adresses IP d'un hôte à rattachements multiples ; dans d'autres cas, le choix peut être fait au sein du logiciel du réseau.

3.3.4.2 Exigences de multi rattachement

Les règles générales suivantes s'appliquent à la sélection d'une adresse IP de source pour l'envoi d'un datagramme à partir d'un hôte à rattachements multiples.

(1) Si le datagramme est envoyé en réponse à un datagramme reçu, l'adresse de source pour la réponse DEVRAIT être l'adresse de destination spécifique de la demande. Voir les paragraphes 4.1.3.5 et 4.2.3.7 et la section "Questions générales" de la [RFC1123] pour les exigences plus spécifiques des couches supérieures.

Autrement, une adresse de source doit être choisie.

(2) Une application DOIT être capable de spécifier explicitement l'adresse de source pour initialiser une connexion ou une demande.

(3) En l'absence d'une telle spécification, le logiciel de réseautage DOIT choisir une adresse de source. Les règles de ce choix sont décrites ci-dessous.

Il y a deux exigences clés qui posent problème en relation avec le rattachement multiple :

(A) Un hôte PEUT éliminer en silence un datagramme entrant dont l'adresse de destination ne correspond pas à l'interface physique par laquelle il a été reçu.

(B) Un hôte PEUT se restreindre lui-même à l'envoi de datagrammes (non acheminés par la source) IP à travers l'interface physique qui correspond à l'adresse IP de source des datagrammes.

Discussion :
Les mises en œuvre d'hôte Internet ont utilisé deux modèles conceptuels différents pour le rattachement multiple, brièvement résumés dans l'exposé suivant. Le présent document ne prend pas position sur le modèle préféré ; chacun semblant avoir sa place. Cette ambivalence est reflétée dans le fait que les questions (A) et (B) sont facultatives.

o Modèle ES fort
Le modèle ES fort (ES, *End System*, c'est-à-dire, hôte) développe la distinction hôte/routeur (ES/IS), et va donc substituer DOIT à PEUT dans les points (A) et (B) ci-dessus. Il tend à modéliser un hôte à rattachement multiple comme un ensemble d'hôtes logiques au sein du même hôte physique.
Par rapport à (A), les défenseurs du modèle ES fort notent que les mécanismes d'acheminement automatique de l'Internet ne pourraient pas acheminer un datagramme vers une interface physique qui ne correspondait pas à l'adresse de destination.
Dans le modèle ES fort, le calcul du chemin d'un datagramme sortant est la transposition :

route(adresse IP de source, adresse IP de destination, TOS) -> routeur

Ici l'adresse de source est incluse comme paramètre afin de choisir un routeur qui soit directement joignable sur l'interface physique correspondante. Noter que ce modèle exige logiquement qu'il y ait en général au moins un routeur par défaut, et de préférence plusieurs, pour chaque adresse IP de source.

o Modèle ES faible
Cette façon de voir ne met pas l'accent sur la distinction ES/IS, et substituera donc NE DOIT PAS à PEUT dans les points (A) et (B). Ce modèle peut être le plus naturel pour les hôtes qui écoutent les protocoles d'acheminement de routeurs, et il est nécessaire pour les hôtes qui ont la fonction de routeur incorporée.

Le modèle ES faible peut causer l'échec du mécanisme Redirect. Si un datagramme est envoyé d'une interface physique qui ne correspond pas à l'adresse de destination, le routeur du premier bond ne va pas réaliser quand il doit envoyer un Redirect. D'un autre côté, si l'hôte a une fonction de routeur incorporée, il a les informations d'acheminement sans avoir à écouter les Redirect.

Dans le modèle ES faible, le calcul de route pour un datagramme sortant est la transposition :

route(adresse IP de destination, TOS) -> routeur, interface

3.3.4.3 Choisir une adresse de source

Discussion :
Lorsqu'elle envoie une demande de connexion initiale (par exemple, un segment TCP "SYN") ou une demande de service de datagramme (par exemple, une interrogation fondée sur UDP) la couche transport sur un hôte à rattachement multiple a besoin de savoir quelle adresse de source utiliser. Si l'application ne le spécifie pas, la couche transport doit demander à la couche IP d'effectuer la transposition conceptuelle :

GET_SRCADDR(adresse IP distante, TOS) -> adresse IP locale

Ici TOS est la valeur du Type-de-Service (voir au paragraphe 3.2.1.6), et le résultat est l'adresse de source désirée. Les règles suivantes sont suggérées pour la mise en œuvre de cette transposition :

(a) Si l'adresse Internet distante est sur un des (sous-)réseaux auxquels l'hôte est directement connecté, une adresse de source correspondante peut être choisie, sauf si l'interface correspondante est connue pour être morte.

(b) L'antémémoire des routes peut être consultée, pour voir si il y a une route active pour le réseau de destination spécifié à travers toute interface réseau ; si il y en a une, on peut choisir une adresse IP locale correspondant à cette interface.

(c) Le tableau des routes statiques, s'il en est un (voir au paragraphe 3.3.1.2) peut de même être consulté.

(d) Les routeurs par défaut peuvent être consultés. Si ces routeurs sont alloués à des interfaces différentes, l'interface correspondant au routeur ayant la plus forte préférence peut être choisie.

À l'avenir, il pourrait y avoir une façon définie pour qu'un hôte à rattachement multiple demande à tous les routeurs de tous les réseaux connectés un avis sur les meilleurs réseaux à utiliser pour une destination donnée.

Mise en œuvre :
On notera que ce processus est essentiellement le même que l'acheminement de datagramme (voir au paragraphe 3.3.1) et donc les hôtes peuvent être capables de combiner la mise en œuvre des deux fonctions.

3.3.5 Retransmission de route de source

Sous réserve des restrictions mentionnées ci-dessous, un hôte PEUT être capable d'agir comme un bond intermédiaire dans une route de source, en transmettant un datagramme acheminé par la source au prochain bond spécifié.

Cependant, en effectuant cette fonction de routeur, l'hôte DOIT ¨respecter toutes les règles pertinentes pour un routeur qui transmet des datagrammes acheminés par la source [RFC1009]. Cela inclut les dispositions spécifiques suivantes, qui subrogent les dispositions d'hôte correspondantes données précédemment dans le présent document:

(A) TTL (voir le paragraphe 3.2.1.7)
Le champ TTL DOIT être décrémenté et le datagramme peut-être éliminé comme spécifié pour un routeur dans [RFC1009].

(B) ICMP Destination injoignable (voir le paragraphe 3.2.2.1)
Un hôte DOIT être capable de générer des messages Destination injoignable avec les codes suivants :
4 (Fragmentation exigée mais DF mis) quand un datagramme acheminé par la source ne peut être fragmenté pour tenir dans le réseau cible;
5 (Échec de route de source) quand un datagramme acheminé par la source ne peut être transmis, par exemple, à cause d'un problème d'acheminement ou parce que le prochain bond d'une route de source stricte n'est pas sur un réseau connecté.

(C) Adresse IP de source (voir le paragraphe 3.2.1.3)
Un datagramme acheminé par la source en cours de transmission PEUT (et normalement doit) avoir une adresse de source qui n'est pas une des adresses IP de l'hôte qui transmet.

(D) Option de route enregistrée (voir le paragraphe 3.2.1.8d)
Un hôte qui transmet un datagramme acheminé par la source contenant une option Route enregistrée DOIT mettre à jour cette option, si il en a la place.

(E) Option Horodatage (voir le paragraphe 3.2.1.8e)
Un hôte qui transmet un datagramme acheminé par la source contenant une option Horodatage DOIT ajouter l'horodatage actuel à cette option, conformément aux règles de cette option.

Pour définir les règles imposant des restrictions à la transmission de datagrammes acheminés par la source par un hôte, on utilise le terme "acheminement par la source local" si le prochain bond doit être à travers la même interface physique que celle par laquelle le datagramme est arrivé ; autrement, c'est un "acheminement par la source non local".

- Il est permis à un hôte d'effectuer sans restriction l'acheminement par la source local.
- Un hôte qui accepte l'acheminement par la source non local DOIT avoir un commutateur configurable pour désactiver la transmission, et ce commutateur DOIT désactiver par défaut.
- L'hôte DOIT satisfaire à toutes les exigences des routeurs sur les filtres de politique configurables [RFC1009] imposant des restrictions à la transmission non locale.

Si un hôte reçoit un datagramme avec une route de source incomplète mais ne le retransmet pas pour une raison quelconque, l'hôte DEVRAIT retourner un message ICMP Destination injoignable (code 5, Échec de route de source) sauf si le datagramme était lui-même un message d'erreur ICMP.

3.3.6 Diffusions

Le paragraphe 3.2.1.3 définit les quatre formes standard d'adresse de diffusion IP :

Diffusion limitée : {-1, -1}
Diffusion dirigée : {<Numéro-de-réseau>,-1}
Diffusion dirigée de sous-réseau : {<Numéro-de-réseau>,<Numéro-de-sous-réseau>,-1}
Diffusion dirigée sur tous les sous-réseaux : {<Numéro-de-réseau>,-1,-1}

Un hôte DOIT reconnaître toutes ces formes dans l'adresse de destination d'un datagramme entrant.

Il y a une classe d'hôtes (4.2BSD Unix et ses dérivés, mais pas 4.3BSD) qui utilise des formes non standard d'adresse de diffusion, en substituant 0 à -1. Tous les hôtes DEVRAIENT reconnaître et accepter toutes ces adresses de diffusion non standard comme adresses de destination d'un datagramme entrant. Un hôte PEUT facultativement avoir une option de

configuration pour choisir la forme 0 ou -1 d'adresse de diffusion, pour chaque interface physique, mais cette option DEVRAIT être par défaut la forme standard (-1).

Lorsque un hôte envoie un datagramme à une adresse de diffusion de couche de liaison, l'adresse de destination IP DOIT être une adresse légale de diffusion IP ou de diffusion groupée IP.

Un hôte DEVRAIT éliminer en silence un datagramme reçu via une diffusion de couche de liaison (voir au paragraphe 2.4) mais ne spécifie pas une adresse de destination IP de diffusion ou de diffusion groupée.

Les hôtes DEVRAIENT utiliser l'adresse de diffusion limitée pour diffuser à un réseau connecté.

Discussion :
L'utilisation de l'adresse de diffusion limitée au lieu de l'adresse de diffusion dirigée peut améliorer la robustesse du système. Des problèmes sont souvent causés par des machines qui ne comprennent pas la pléthore d'adresses de diffusion (voir au paragraphe 3.2.1.3) ou qui ont des idées différentes sur les adresses de diffusion qui sont utilisées. Le principal exemple de ce dernier cas est celui de machines qui ne comprennent pas le sous réseautage mais sont rattachées à un réseau subdivisé en sous-réseaux. L'envoi d'une diffusion de sous-réseau pour le réseau connecté va perturber ces machines, qui vont la voir comme un message destiné à un autre hôte.

Il y a eu des discussions sur le fait de savoir si un datagramme adressé à l'adresse de diffusion limitée devrait être envoyé de toutes les interfaces d'un hôte à rattachements multiples. La présente spécification ne prend pas position sur cette question.

3.3.7 Diffusion groupée sur IP

Un hôte DEVRAIT prendre en charge la diffusion groupée IP locale sur tous les réseaux connectés pour lesquels une transposition des adresses IP de classe D en adresses de couche de liaison a été spécifiée (voir ci-dessous). La prise en charge de la diffusion groupée IP locale inclut l'envoi de datagrammes en diffusion groupée, l'adhésion à des groupes de diffusion groupée et la réception de datagrammes en diffusion groupée, ainsi que de quitter les groupes de diffusion. Cela implique la prise en charge de tout [RFC1112] sauf le protocole IGMP lui-même, qui est FACULTATIF.

Discussion :
IGMP fournit des routeurs qui sont capables d'acheminement de diffusion groupée avec les informations requises pour prendre en charge la diffusion groupée IP à travers plusieurs réseaux. À l'heure actuelle, les routeurs à acheminement en diffusion groupée sont à l'état expérimental et ne sont pas largement disponibles. Pour les hôtes qui ne sont pas connectés aux réseaux avec des routeurs à acheminement en diffusion groupée ou qui n'ont pas besoin de recevoir des datagrammes en diffusion groupée provenant d'autres réseaux, IGMP ne sert à rien et est donc facultatif pour l'instant. Cependant, le reste de la [RFC1112] est actuellement recommandé pour pouvoir fournir l'accès de couche IP à l'adressage en diffusion groupée de réseau local, comme solution de remplacement préférable à l'adressage de diffusion local. Il est prévu que IGMP deviendra recommandé à l'avenir, lorsque les routeurs à acheminement en diffusion groupée seront plus largement disponibles.

Si IGMP n'est pas mis en œuvre, un hôte DEVRAIT quand même se joindre au groupe "tous-les-hôtes" (224.0.0.1) lorsque la couche IP est initialisée et en rester membre aussi longtemps que la couche IP est active.

Discussion :
Se joindre au groupe "tous-les-hôtes" prendra strictement en charge les utilisations locales de diffusion groupée, par exemple, un protocole de découverte des routeurs, même si IGMP n'est pas mis en œuvre.

La transposition des adresses IP de classe D en adresses locales est actuellement spécifiée pour les types de réseaux suivants :

- Ethernet/IEEE 802.3, tel que défini dans la [RFC1112].
- Tout réseau qui accepte l'adressage en diffusion mais pas en diffusion groupée : toutes les adresses IP de classe D se transposent en adresse de diffusion locale.
- Tout type de liaison point à point (par exemple, des liaisons SLIP ou HDLC) : pas de transposition nécessaire. Tous les datagrammes IP en diffusion groupée sont envoyés tels quels, au sein du tramage local.

Les transpositions pour les autres types de réseaux seront spécifiées à l'avenir.

Un hôte DEVRAIT fournir un moyen pour que les protocoles ou applications de couche supérieure déterminent lequel des réseaux connectés de l'hôte prend en charge l'adressage IP en diffusion groupée.

3.3.8 Rapport d'erreurs

Chaque fois que c'est praticable, les hôtes DOIVENT retourner des datagrammes d'erreur ICMP lorsqu'ils détectent une erreur, excepté dans les cas où le retour d'un message d'erreur ICMP est spécifiquement interdit.

Discussion :
Un phénomène courant dans les réseaux de datagrammes est le "syndrome du trou noir" : les datagrammes sont envoyés, mais rien ne revient. Sans datagramme d'erreur, il est difficile à l'utilisateur de se rendre compte de la nature du problème.

3.4 Interface de la couche Internet/Transport

L'interface entre la couche IP et la couche transport DOIT fournir le plein accès à tous les mécanismes de la couche IP, y compris les options, Type-de-Service, et Durée-de-vie. La couche transport DOIT avoir des mécanismes pour établir ces paramètres d'interface, ou fournir un chemin pour les passer à partir d'une application, ou les deux.

Discussion :
Les applications sont vivement invitées à faire usage de ces mécanismes là où ils sont applicables, même lorsque les mécanismes ne sont pas actuellement efficaces dans l'Internet (par exemple, le TOS). Cela permettra que ces mécanismes soient immédiatement utiles lorsqu'ils deviendront efficaces, sans gros efforts de remise à niveau du logiciel d'hôte.

Nous décrivons maintenant une interface conceptuelle entre la couche transport et la couche IP, comme un ensemble d'appels de procédure. Ceci est une extension des informations du paragraphe 3.3 de la [RFC0791].

* Datagramme Send

  SEND(src, dst, prot, TOS, TTL, BufPTR, len, Id, DF, opt => result)

  où les paramètres sont définis dans la RFC-791. Passer un paramètre Id est facultatif ; voir au paragraphe 3.2.1.5.

* Datagramme Receive

  RECV(BufPTR, prot => result, src, dst, SpecDest, TOS, len, opt)

  Tous les paramètres sont définis dans la RFC-791, excepté :

  SpecDest = adresse spécifique de destination de datagramme (définie au paragraphe 3.2.1.3)

  Le paramètre dst résultant contient l'adresse de destination du datagramme. Comme cela peut être une adresse de diffusion ou de diffusion groupée, le paramètre SpecDest (qui ne figure pas dans la RFC-791) DOIT être passé. Le paramètre opt contient toutes les options IP reçues dans le datagramme ; celles-ci DOIVENT aussi être passées à la couche transport.

* Choix de l'adresse de source

  GET_SRCADDR(remote, TOS) -> local

  remote = adresse IP distante
  TOS = Type de service
  local = adresse IP locale

  Voir au paragraphe 3.3.4.3.

* Trouver les tailles maximum de datagramme

  GET_MAXSIZES(local, remote, TOS) -> MMS_R, MMS_S

  MMS_R = taille maximum de message de transport reçu.

MMS_S = taille maximum de message de transport envoyé.
(local, remote, TOS sont définis ci-dessus)

Voir les paragraphes 3.3.2 et 3.3.3.

* Avis de livraison réussie

ADVISE_DELIVPROB(sense, local, remote, TOS)

Ici le paramètre sense est un fanion d'un bit indiquant si un avis positif ou négatif est donné ; voir l'exposé du paragraphe 3.3.1.4. Les autres paramètres ont été définis précédemment.

* Message ICMP envoyé

SEND_ICMP(src, dst, TOS, TTL, BufPTR, len, Id, DF, opt) -> result

(Paramètres définis dans la RFC-791).

Passer un paramètre Id est facultatif ; voir au paragraphe 3.2.1.5. La couche transport DOIT être capable d'envoyer certains messages ICMP : Accès injoignable ou n'importe lequel des messages de type interrogation. Cette fonction pourrait être considérée comme un cas particulier de l'appel SEND(), bien sûr ; on le décrit à part pour être clair.

* Message ICMP reçu

RECV_ICMP(BufPTR) -> result, src, dst, len, opt

(Paramètres définis dans la RFC-791).

La couche IP DOIT remonter certains messages ICMP au logiciel approprié de couche transport. Cette fonction pourrait être considérée comme un cas particulier de l'appel RECV(), bien sûr ; on le décrit à part pour être clair.

Pour un message d'erreur ICMP, les données qui sont remontées DOIVENT inclure l'en-tête Internet d'origine plus tous les octets du message d'origine qui sont inclus dans le message ICMP. Ces données seront utilisées par la couche transport pour localiser les informations d'état de connexion, s'il en est.

En particulier, les messages ICMP suivants sont à remonter :
- o Destination injoignable
- o Extinction de source
- o Réponse d'écho (à l'interface d'utilisateur ICMP, sauf si la Demande d'écho a été générée dans la couche IP)
- o Réponse d'horodatage (à l'interface d'utilisateur ICMP)
- o Délai dépassé

Discussion :
À l'avenir, il pourra y avoir des ajouts à cette interface pour passer des données de chemin (voir au paragraphe 3.3.1.3) entre les couches IP et transport.

3.5 Résumé des exigences de la couche Internet

Caractéristique	Paragr.	DOIT	DEVRAIT	PEUT	NE DEVRAIT PAS	NE DOIT PAS	Note
Met en œuvre IP et ICMP	3.1	x					
Traite le rattach. mult. distant dans la couche appli.	3.1	x					
Prend en charge le rattachement multiple local	3.1			x			
Satisfait aux ex. des routeurs s'il transmet des dtgms	3.1	x					
Commutation de config. pour les routeurs incorporés	3.1	x					1
Commutation de config par défaut à non-routeur	3.1	x					1
Auto-config fondée sur le nombre d'interfaces	3.1					x	1
Capable d'enregistrer les datagrammes éliminés	3.1		x				

Compteur d'enregistrement	3.1		x				
Éliminer en silence Version != 4	3.2.1.1	x					
Vérif. la s. de ctrl IP, élim. en s. mauvais datagramme	3.2.1.2	x					
Adressage :							
Adressage de sous-réseau (RFC-950)	3.2.1.3	x					
Adr. de source doit être la propre adresse IP de l'hôte	3.2.1.3	x					
Élim. en s. dtgm avec mauvaise adr. de destination	3.2.1.3	x					
Élim. en s. dtgm avec mauvaise adresse de source	3.2.1.3	x					
Prise en charge du réassemblage	3.2.1.4	x					
Conserver le même champ Id dans un dtgm identique	3.2.1.5			x			
TOS :							
Permettre à la couche de transport d'établir le TOS	3.2.1.6	x					
Passer le TOS reçu à la couche transport	3.2.1.6		x				
Utiliser les transpo. de CL RFC-795 pour le TOS	3.2.1.6				x		
TTL :							
Envoi de paquet avec un TTL de 0	3.2.1.7					x	
Éliminer les paquets reçus avec TTL < 2	3.2.1.7					x	
Permettre à la couche transport d'établir le TTL	3.2.1.7	x					
Le TTL fixé est configurable	3.2.1.7	x					
Options IP :							
Permettre à la c. transport d'envoyer les options IP	3.2.1.8	x					
Passer toutes les options IP reçues à la c. supérieure	3.2.1.8	x					
La couche IP ignore en silence les options inconnues	3.2.1.8	x					
Option de sécurité	3.2.1.8a			x			
Option d'identifiant du flux d'envoi	3.2.1.8b				x		
Ignorer en silence l'option d'identifiant de flux	3.2.1.8b	x					
Option Enregistrement du chemin	3.2.1.8d			x			
Option Horodatage	3.2.1.8e			x			
Option Route de source :							
Options Route de source d'origine & terminaison	3.2.1.8c	x					
Datagramme avec SR complétée passé à la CT	3.2.1.8c	x					
Construire un ch. de retour correct (non redondant)	3.2.1.8c	x					
Envoi de plusieurs options SR dans un en-tête	3.2.1.8c					x	
ICMP :							
Éliminer en silence le msg ICMP de type inconnu	3.2.2	x					
Inclure plus de 8 octets du datagramme d'origine	3.2.2			x			
Inclure les octets tels que reçus	3.2.2	x					
Démux Erreur ICMP au protocole de transport	3.2.2	x					
Envoi de message d'erreur ICMP avec TOS=0	3.2.2		x				
Envoi de message d'erreur ICMP pour :							
- message d'erreur ICMP	3.2.2					x	
- diffusion IP ou diff groupée IP	3.2.2					x	
- diffusion de couche de liaison	3.2.2					x	
- fragment non initial	3.2.2					x	
- datagramme avec adresse de src non unique	3.2.2					x	
Retour de msgs d'erreur ICMP (si non interdit)	3.3.8	x					
Destination injoignable :							
Générer Dest. injoignable (code 2/3)	3.2.2.1		x				
Passer Dest injoignable ICMP à la couche sup.	3.2.2.1	x					
Couche sup. agit sur Dest injoignable	3.2.2.1		x				
Interprète Dest injoignable comme simple conseil	3.2.2.1	x					
Rediriger :							
L'hôte envoie Rediriger	3.2.2.2				x		
Màj mém. cache des routes à réception de Rediriger	3.2.2.2	x					
Traite les Rediriger d'hôte et de réseau	3.2.2.2	x					
Éliminer le Rediriger illégal	3.2.2.2		x				
Source éteinte :							
Envoi de Source éteinte si mémoire tampon pleine	3.2.2.3			x			

Passer Source éteinte à la couche sup.	3.2.2.3	x					
Action de couche sup sur Source éteinte	3.2.2.3		x				
Délai excédé : passé à la couche sup.	3.2.2.4	x					
Problème de paramètre :							
Envoi des msgs Problème de paramètre	3.2.2.5		x				
Passer Problème de paramètre à la couche sup.	3.2.2.5	x					
Rapport de Problème de paramètre à l'usager	3.2.2.5			x			
Demande ou Réponse d'écho ICMP :							
Écho serveur et écho client	3.2.2.6	x					
Écho client	3.2.2.6		x				
Éliminer demande d'écho à une adresse de diff.	3.2.2.6			x			
Éliminer demande d'écho à une adresse de diff. gr.	3.2.2.6			x			
Utilise une adr. spéc. de dest. comme src rép. écho	3.2.2.6	x					
Envoyer les mêmes données dans la rép. d'écho	3.2.2.6	x					
Passer la réponse d'écho à la couche sup.	3.2.2.6	x					
Reflet des options Route enreg., Horodatage	3.2.2.6		x				
Inverser et refléter l'option Route de source	3.2.2.6	x					
Demande ou Rép. d'information ICMP :	3.2.2.7				x		
Horodatage et Rép d'horod. ICMP :	3.2.2.8			x			
Minimiser la variabilité du délai	3.2.2.8		x				1
Éliminer en silence l'horodatage en diff.	3.2.2.8			x			1
Éliminer en silence l'horodatage en diff. group.	3.2.2.8			x			1
Util. adr. spécif. de dest comme src de Rép. horod.	3.2.2.8	x					1
Reflet des options Route enreg., Horodatage	3.2.2.6		x				1
Inverser et refléter l'option Route de source	3.2.2.8	x					1
Passer la Réponse d'horodat. à la couche sup.	3.2.2.8	x					1
Respect des règles de "valeur standard"	3.2.2.8	x					1
Demande et réponse de gabarit d'adresse ICMP :							
Source de gabarit d'adresse configurable	3.2.2.9	x					
Prise en charge de la config. statique de gab. d'adr.	3.2.2.9	x					
Obtention dynamique du gab. d'adr à l'amorçage	3.2.2.9			x			
Obtention d'adr. via Dde/Rép. de gab. adr. ICMP	3.2.2.9			x			
Réémission de Demande de gab d'adr sans réponse	3.2.2.9	x					3
Supposer le gabarit par défaut sans réponse	3.2.2.9		x				3
Màj de gab d'adr seulement pour 1ère Réponse	3.2.2.9	x					3
Vérif. de vraisembl. sur gabarit d'adresse	3.2.2.9		x				
Envoi de msgs Réponse de gab. d'adr. non autorisés	3.2.2.9					x	
Explicitement configurés pour être agent	3.2.2.9	x					
Config statique => fanion autor-gabarit-adresse	3.2.2.9		x				
Diff. de Rép de gabarit d'adresse à l'init.	3.2.2.9	x					3
Routage des datagrammes sortants :							
Utilise gab. d'adr. dans la décision locale/distante	3.3.1.1	x					
Fonctionne sans routeur sur le réseau connecté	3.3.1.1	x					
Entretien de la "antémémoire des routes" des routeurs du prochain bond	3.3.1.2	x					
Même traitement pour Rediriger d'hôte et réseau	3.3.1.2		x				
Sans entrée d'antémémoire, utilise routeur par défaut	3.3.1.2	x					
Accepte plusieurs routeurs par défaut	3.3.1.2	x					
Fournit le tableau des routes statiques	3.3.1.2			x			
Fanion : route subrogée par les Rediriger	3.3.1.2			x			
Antémémoire de route sur hôte, pas adresse réseau	3.3.1.3			x			
Inclure le TOS dans l'antémémoire de route	3.3.1.3		x				
Capable détecter défaillance routeur du proch. bond	3.3.1.4	x					
Suppose que la route est toujours bonne	3.3.1.4				x		
Envoie des ping aux routeurs en permanence	3.3.1.4					x	
Ne ping que lors de l'envoi de trafic	3.3.1.4	x					
Ne ping que quand il n'y a pas d'indication positive	3.3.1.4	x					
Les couches inférieure et supérieure donnent l'avis	3.3.1.4		x				

Passer d'un routeur par défaut défaillant à un autre	3.3.1.5	x					
Méthode manuelle d'entrée des infos de config.	3.3.1.6	x					
Réassemblage et fragmentation :							
Capable de réassembler les datagrammes entrants	3.3.2	x					
Au moins 576 octets par datagramme	3.3.2	x					
EMTU_R configurable ou indéfini	3.3.2		x				
Couche transport capable d'apprendre MMS_R	3.3.2	x					
Envoi ICMP Délai dépassé sur tempo. de réass.	3.3.2	x					
Valeur de tempo de réassemblage fixée	3.3.2		x				
Passe MMS_S aux couches supérieures	3.3.3	x					
Fragmentation locale des paquets sortants	3.3.3			x			
Autrement n'envoie pas plus que MMS_S	3.3.3	x					
Envoi de max 576 à une destination hors réseau	3.3.3		x				
Fanion de config. MTU-tous-sous-réseaux	3.3.3			x			
Rattachement multiple :							
Répond avec la même adr que adr spécif de dest.	3.3.4.2		x				
Permet à l'application de choisir l'adr IP locale	3.3.4.2	x					
Élim. en silence datagram sur "mauvaise" interface	3.3.4.2			x			
N'envoie datagram que sur la "bonne" interface	3.3.4.2			x			4
Transmission de route de source :							
Transmet le datagramme avec option route de source	3.3.5			x			1
Respecte les règles de routeur correspondantes	3.3.5	x					1
Met à jour TTL selon les règles des routeurs	3.3.5	x					1
Capable de générer les codes 4, 5 d'erreur ICMP	3.3.5	x					1
Adr IP de src pas sur l'hôte local	3.3.5			x			1
Mise à jour options Horodatage, Route enregist.	3.3.5	x					1
Commut configurable pour SRing non local	3.3.5	x					1
Désactivé par défaut	3.3.5	x					1
Respect règles d'accès de rtr pour SRing non local	3.3.5	x					1
Si non transmis, envoi de Dest injoign. (code 5)	3.3.5		x				2
Diffusion :							
Adresse de diff comme adresse de source IP	3.2.1.3					x	
Reçoit les formats de diff 0 ou -1	3.3.6		x				
Option configurable pour envoi de diffusion 0 ou -1	3.3.6			x			
Diffusion -1 par défaut	3.3.6		x				
Reconnaît tous les formats d'adresse de diffusion	3.3.6	x					
Utilise adr IP diff/diff group dans diff couche liaison	3.3.6	x					
Éliminer en silence dgms en diff couche liaison seule	3.3.6		x				
Utilise adr diff limitée pour réseau connecté	3.3.6		x				
Diffusion groupée :							
Accepte diff group IP locale (RFC-1112)	3.3.7		x				
Accepte IGMP (RFC-1112)	3.3.7			x			
Se joint au groupe tous-hôtes au démarrage	3.3.7		x				
Couches sup apprennent capacité diff group d'interf.	3.3.7		x				
Interface :							
Permet à la couche transport d'utiliser tous mécan. IP	3.4	x					
Passer ID d'interface à couche transport	3.4	x					
Passe toutes options IP à couche transport	3.4	x					
Couche transport peut envoyer certains msgs ICMP	3.4	x					
Passe msgs ICMP spécifiés à couche transport	3.4	x					
Inclure en-tête IP +8 octets ou plus de l'origine.	3.4	x					
Capable sauter grande construct. en une seule fois	3.4		x				

Notes :
(1) Seulement si la caractéristique est mise en œuvre
(2) Cette exigence est subrogée si le datagramme est un message d'erreur ICMP.
(3) Seulement si la caractéristique est mise en œuvre et est activée par la configuration.
(4) Qu'avec la fonction de routeur incorporée ou si acheminé par la source.

4 Protocoles de transport

4.1 Protocole des datagrammes d'utilisateur -- UDP

4.1.1 Introduction

Le protocole de datagramme d'utilisateur (UDP, *User Datagram Protocol*) [RFC0768] n'offre qu'un service de transport minimal – livraison de datagramme non garantie – et donne aux applications un accès direct au service de datagramme de la couche IP. UDP est utilisé par des applications qui n'exigent pas le niveau de service de TCP ou qui souhaitent utiliser des services de communications (par exemple, livraison en diffusion ou diffusion groupée) non disponibles à partir de TCP.

UDP est presque un protocole nul ; le seul service qu'il fournisse sur IP est la somme de contrôle des données et le multiplexage par numéro d'accès. Donc, tout programme d'application fonctionnant sur UDP doit traiter directement les problèmes de communication de bout en bout qu'aurait traité un protocole de mode connexion -- par exemple, la retransmission pour une livraison fiable, la mise en paquet et le réassemblage, le contrôle des flux, le règlement de l'encombrement, etc., lorsque ceux-ci sont nécessaires. Le couplage assez complexe entre IP et TCP sera reflété par le couplage entre UDP et de nombreuses applications qui utilisent UDP.

4.1.2 Revue du protocole

Il n'y a pas d'erreurs connues dans la spécification d'UDP.

4.1.3 Questions spécifiques

4.1.3.1 Accès

Les accès bien connus UDP suivent les mêmes règles que les accès TCP bien connus ; voir au paragraphe 4.2.2.1 ci-dessous.

Si un datagramme arrive adressé à un accès UDP pour lequel il n'y a pas d'appel LISTEN en cours, UDP DEVRAIT envoyer un message ICMP accès injoignable.

4.1.3.2 Options IP

UDP DOIT passer toute option IP qu'il reçoit de la couche IP de façon transparente à la couche d'application.

Une application DOIT être capable de spécifier les options IP à envoyer dans ses datagrammes UDP, et UDP DOIT passer ces options à la couche IP.

Discussion :
À présent, les seules options qui doivent être passées à travers UDP sont Route de source, Route enregistrée, et Horodatage. Cependant, de nouvelles options pourraient être définies à l'avenir, et UDP n'a pas besoin et ne devrait pas faire d'hypothèses sur le format ou le contenu des options qu'il passe de ou vers l'application ; une exception à cela pourrait être celle d'une option de sécurité de couche IP.

Une application fondée sur UDP aura besoin d'obtenir une route de source d'un datagramme de demande et de fournir une route inverse pour l'envoi de la réponse correspondante.

4.1.3.3 Messages ICMP

UDP DOIT passer à la couche d'application tous les messages d'erreur ICMP qu'il reçoit de la couche IP. Conceptuellement au moins, cela peut être réalisé par un appel à la routine ERROR_REPORT (voir au paragraphe 4.2.4.1).

Discussion :
Noter que les messages d'erreur ICMP résultant de l'envoi d'un datagramme UDP sont reçus en asynchrone. Une application fondée sur UDP qui veut recevoir les messages d'erreur ICMP est responsable de la maintenance de l'état nécessaire au démultiplexage de ces messages quand ils arrivent ; par exemple, l'application peut garder une opération de

réception active à cette fin. L'application est aussi responsable d'éviter la confusion provenant d'un message d'erreur ICMP retardé résultant d'un usage antérieur du même ou des mêmes accès.

4.1.3.4 Sommes de contrôle UDP

Un hôte DOIT mettre en œuvre la facilité de générer et valider les sommes de contrôle UDP. Une application PEUT facultativement être capable de contrôler si une somme de contrôle UDP sera générée, mais il DOIT par défaut avoir la somme de contrôle activée.

Si un datagramme UDP est reçu avec une somme de contrôle qui est différente de zéro et invalide, UDP DOIT éliminer en silence le datagramme. Une application PEUT facultativement être capable de contrôler si les datagrammes UDP sans somme de contrôle devraient être éliminés ou passés à l'application.

Discussion :
Certaines applications qui fonctionnent normalement seulement à travers des réseaux de zone locale ont choisi de désactiver les sommes de contrôle UDP pour des raisons d'efficacité. Il en résulte que de nombreux cas d'erreurs non détectées ont été rapportés. La question de savoir s'il peut être conseillé de désactiver le mécanisme de somme de contrôle de UDP est très controversée.

Mise en œuvre :
Il existe une erreur courante dans la mise en œuvre des sommes de contrôle UDP. À la différence des sommes de contrôle TCP, la somme de contrôle UDP est facultative ; la valeur zéro est transmise dans le champ de somme de contrôle d'un en-tête UDP pour indiquer l'absence de somme de contrôle. Si l'émetteur calcule réellement une somme de contrôle UDP de zéro, il doit transmettre la somme de contrôle toute de uns (65535). Aucune action particulière n'est requise du receveur, car zéro et 65535 sont équivalents en arithmétique de compléments à un.

4.1.3.5 Multi rattachement UDP

Lors de la réception d'un datagramme UDP, son adresse spécifique de destination DOIT être remontée à la couche d'application.

Un programme d'application DOIT être capable de spécifier l'adresse IP de source à utiliser pour l'envoi d'un datagramme UDP ou de le laisser non spécifié (auquel cas le logiciel de réseautage choisira une adresse de source appropriée). Il DEVRAIT y avoir un moyen de communiquer l'adresse de source choisie jusqu'à la couche d'application (par exemple, de telle sorte que l'application puisse ultérieurement recevoir un datagramme de réponse de la seule interface correspondante).

Discussion :
Une application de demande/réponse qui utilise UDP devrait utiliser pour la réponse une adresse de source qui soit la même que l'adresse spécifique de destination de la demande. Voir la section "Questions générales" de la [RFC1123].

4.1.3.6 Adresses invalides

Un datagramme UDP reçu avec une adresse IP de source invalide (par exemple, une adresse de diffusion ou de diffusion groupée) doit être éliminé par UDP ou par la couche IP (voir au paragraphe 3.2.1.3).

Lorsqu'un hôte envoie un datagramme UDP, l'adresse de source DOIT être une des adresses IP de l'hôte.

4.1.4 Interface de couche UDP/Application

L'interface d'application à UDP DOIT fournir les services complets de l'interface IP/transport décrits au paragraphe 3.4 du présent document. Et donc, une application utilisant UDP a besoin des fonctions des commandes GET_SRCADDR(), GET_MAXSIZES(), ADVISE_DELIVPROB(), et RECV_ICMP() décrits au paragraphe 3.4. Par exemple, GET_MAXSIZES() peut être utilisé pour acquérir la taille maximum de datagramme UDP effective maximum pour un triplet {interface,hôte-distant,TOS} particulier.

Un programme de couche d'application DOIT être capable d'établir les valeurs de TTL et de TOS aussi bien que des options IP pour l'envoi d'un datagramme UDP, et ces valeurs doivent être passées de façon transparente à la couche IP. UDP PEUT passer le TOS reçu à la couche d'application.

4.1.5 Résumé des exigences pour UDP

Caractéristique	Paragraphe	Doit	Devrait	Peut
UDP				
Envoi UDP de Accès injoignable	4.1.3.1		x	
Options IP dans UDP				
- Passer options IP reçues à couche d'application	4.1.3.2	x		
- Couche d'application peut spécifier les options IP dans l'envoi	4.1.3.2	x		
- UDP passe les options IP à la couche IP	4.1.3.2	x		
Passe les messages ICMP à la couche d'application	4.1.3.3	x		
Somme de contrôle UDP :				
- Capable de générer/vérifier la somme de contrôle	4.1.3.4	x		
- Éliminer en silence une mauvaise somme de contrôle	4.1.3.4	x		
- Option de l'envoyeur de ne pas générer la somme de contrôle	4.1.3.4			x
- La somme de contrôle est la valeur par défaut	4.1.3.4	x		
- Option du receveur d'exiger la somme de contrôle	4.1.3.4			x
Rattachement multiple UDP :				
- Passer l'adresse spécifique de dest. à l'application	4.1.3.5	x		
- La couche d'applic peut spécifier l'adresse IP locale	4.1.3.5	x		
- La couche d'applic spécifie l'adresse IP locale brute	4.1.3.5	x		
- La couche d'applic est notifiée de l'adr. IP loc. utilisée	4.1.3.5		x	
Mauvaise adr IP de src éliminée en silence par UDP/IP	4.1.3.6	x		
Envoi seulement d'adresses IP de source valides	4.1.3.6	x		
Services d'interface d'application UDP				
Interface IP complète du § 3.4 pour l'application	4.1.4	x		
- Capable de spécifier les options TTL, TOS, IP dans l'envoi de datagramme	4.1.4	x		
- Passe le TOS reçu à la couche d'application	4.1.4			x

4.2 Protocole de commande de transmission -- TCP

4.2.1 Introduction

Le protocole de commande de transmission (TCP, *Transmission Control Protocol* [RFC0793] est le principal protocole de transport par circuit virtuel pour la série des protocoles de l'Internet. TCP fournit une livraison fiable, en séquence, de flux d'octets en bilatéral (octets de 8 bits). TCP est utilisé par les applications qui ont besoin d'un service de transport orienté connexion fiable, par exemple, la messagerie (SMTP), le transfert de fichiers (FTP), et le service de terminal virtuel (Telnet) ; les exigences pour ces protocoles de couche d'application sont décrites dans [RFC1123].

4.2.2 Découverte du protocole

4.2.2.1 Les accès bien connus : RFC-793, paragraphe 2.7

Discussion :
TCP réserve les numéros d'accès dans la gamme 0 à 255 pour les accès "bien connus", utilisés pour accéder aux services qui sont normalisés à travers l'Internet. Le reste de l'espace des accès peut être librement alloué aux processus d'application. Les définitions des accès bien connus actuels figurent dans la RFC intitulée "Numéros alloués" [RFC1700]. Un pré-requis pour la définition de nouveaux accès bien connus est qu'une RFC documente le service proposé avec suffisamment de détails pour permettre de nouvelles mises en œuvre.

Certains systèmes étendent cette notion en ajoutant une troisième subdivision de l'espace des accès TCP : les accès réservés, qui sont généralement utilisés pour des services spécifiques du système d'exploitation. Par exemple, les accès réservés peuvent tomber entre 256 et une limite supérieure dépendant du système. Certains systèmes choisissent en plus de protéger les accès bien connus et réservés en ne permettant qu'à des utilisateurs privilégiés d'ouvrir des connexions TCP avec ces valeurs d'accès. Ceci est parfaitement raisonnable tant que l'hôte ne suppose pas que tous les hôtes protègent leurs accès à faible numéros de cette façon.

4.2.2.2 Utilisation de Push : RFC-793 paragraphe 2.8

Lorsque une application produit une série d'appels SEND sans établir le fanion PUSH, TCP PEUT agréger les données en interne sans les envoyer. De même, lorsque une série de segments est reçue sans le bit PSH, TCP PEUT mettre les données en file d'attente en interne sans les passer à l'application receveuse.

Le bit PSH n'est pas un marqueur d'enregistrement et il est indépendant des frontières de segment. L'émetteur DEVRAIT compacter les bits PSH successifs lorsqu'il met les données en paquet, pour envoyer le segment le plus grand possible.

TCP PEUT mettre en œuvre les fanions PUSH sur les appels SEND. Si les fanions PUSH ne sont pas mis en œuvre, le TCP d'envoi : (1) ne doit pas mettre indéfiniment en mémoire tampon les données, et (2) DOIT mettre le bit PSH dans le dernier mis en mémoire tampon (c'est-à-dire, lorsque il n'y a plus de données à envoyer dans la file d'attente).

La discussion des pages 48, 50 et 74 de la RFC-793 implique à tort que le fanion PSH reçu doit être passé à la couche d'application. Passer un fanion PSH reçu à la couche d'application est maintenant FACULTATIF.

Un programme d'application est logiquement obligé d'établir le fanion PUSH dans un appel SEND chaque fois qu'il a besoin de forcer la livraison des données pour éviter une situation insoluble de communication. Cependant, TCP DEVRAIT envoyer un segment de la taille maximum chaque fois que possible, pour améliorer les performances (voir au paragraphe 4.2.3.4).

Discussion :
Lorsque le fanion PUSH n'est pas mis en œuvre sur les appels SEND, c'est-à-dire, lorsque les interfaces application/TCP utilisent un modèle de temps réel pur, la responsabilité de l'agrégation de tous les petits fragments de données pour former des segments de taille raisonnable incombe partiellement à la couche d'application.

Généralement, un protocole d'application interactif doit établir le fanion PUSH au moins dans le dernier appel SEND dans chaque séquence de commande ou réponse. Un protocole de transfert en vrac comme FTP devrait établir le fanion PUSH sur le dernier segment d'un fichier ou lorsque c'est nécessaire pour empêcher une situation insoluble de mémoire tampon.

Chez le receveur, le bit PSH force la livraison des données en mémoire tampon à l'application (même si moins d'une pleine mémoire tampon a été reçue). À l'inverse, l'absence du bit PSH peut être utilisée pour éviter des appels de réveil inutiles au processus d'application ; ceci peut être une importante optimisation des performances pour de grands hôtes en temps partagé. Passer le bit PSH à l'application réceptrice permet une optimisation analogue au sein de l'application.

4.2.2.3 Taille de fenêtre : RFC-793 paragraphe 3.1

La taille de fenêtre DOIT être traitée comme un nombre non signé, ou alors les fenêtres de grande taille apparaîtront comme des fenêtres négatives et TCP ne fonctionnera pas. Il est RECOMMANDÉ que les mises en œuvre réservent des champs de 32 bits pour les tailles de fenêtre d'envoi et de réception dans l'enregistrement de connexion et de faire tous les calculs de fenêtre avec 32 bits.

Discussion :
On sait que le champ de fenêtre dans l'en-tête TCP est trop petit pour les chemins à grande vitesse et fort délai. Des options TCP expérimentales ont été définies pour étendre la taille de fenêtre ; voir par exemple la [RFC1072]. En anticipation de l'adoption d'une telle extension, les mises en œuvre de TCP devraient traiter les fenêtres comme ayant 32 bits.

4.2.2.4 Pointeur d'urgence : RFC-793 paragraphe 3.1

La seconde phrase est erronée : le pointeur d'urgence pointe sur le numéro de séquence du DERNIER octet (pas le DERNIER+1) dans une séquence de données urgentes. La description de la page 56 (dernière phrase) est correcte.

TCP DOIT prendre en charge une séquence de données urgentes de n'importe quelle longueur.

TCP DOIT informer la couche d'application en asynchrone chaque fois qu'elle reçoit un pointeur Urgent et qu'il n'y avait pas précédemment de données urgentes en cours, ou chaque fois que le pointeur Urgent avance dans le flux des données. Il DOIT y avoir un moyen pour que l'application apprenne combien il reste de données urgentes à lire sur la connexion, ou au moins de déterminer si il reste des données urgentes à lire ou non.

Discussion : Bien que le mécanisme Urgent puisse être utilisé pour n'importe quelle application, il est normalement utilisé pour envoyer des commandes de type "interrupt"- à un programme Telnet (voir la section "Utiliser une séquence Telnet Synch" dans [RFC1123]).

La notification asynchrone ou "hors bande" permet à l'application de passer en "mode urgent", dans la lecture des données sur la connexion TCP. Cela permet d'envoyer les commandes de contrôle à une application dont les mémoires tampon d'entrée normales sont pleines de données non traitées.

Mise en œuvre :
L'appel générique ERROR-REPORT() décrit au paragraphe 4.2.4.1 est un mécanisme possible pour informer l'application de l'arrivée de données urgentes.

4.2.2.5 Options TCP : RFC-793 paragraphe 3.1

TCP DOIT être capable de recevoir une option TCP dans tout segment. TCP DOIT ignorer sans erreur toute option TCP qu'il ne met pas en œuvre, en supposant que l'option a un champ Longueur (toutes les options TCP définies à l'avenir auront des champs Longueur). TCP DOIT être prêt à traiter une longueur d'option illégale (par exemple, zéro) sans défaillance ; une suggestion de procédure est de réinitialiser la connexion et d'enregistrer la cause.

4.2.2.6 Option Taille de segment maximum : RFC-793 paragraphe 3.1

TCP DOIT mettre en œuvre l'option Taille de segment maximum (MSS, *Maximum Segment Size*) à la fois en émission et en réception [RFC0879].

TCP DEVRAIT envoyer une option MSS dans chaque segment SYN lorsque sa MSS reçue diffère du 536 par défaut, et PEUT toujours l'envoyer.

Si une option MSS n'est pas reçue à l'établissement de la connexion, TCP DOIT supposer une MSS d'envoi par défaut de 536 (576-40) [RFC0879].

La taille maximum d'un segment qu'envoie réellement TCP, la "MSS d'envoi effective," DOIT être la taille la plus petite entre la MSS d'envoi (qui reflète la taille de mémoire tampon de réassemblage disponible chez l'hôte distant) et la plus grande taille permise par la couche IP :

Eff.snd.MSS = min(SendMSS+20, MMS_S) - TCPhdrsize - IPoptionsize

où :
* SendMSS est la valeur de MSS reçue de l'hôte distant, ou le 536 par défaut si aucune option MSS n'est reçue.
* MMS_S est la taille maximum de message de couche transport que TCP puisse envoyer.
* TCPhdrsize est la taille de l'en-tête TCP ; c'est normalement 20, mais peut être plus grande si les options TCP sont à envoyer.
* IPoptionsize est la taille de toute option IP que TCP va passer à la couche IP ave le message en cours.

La valeur MSS à envoyer dans une option MSS doit être inférieure ou égale à :

MMS_R - 20

où MMS_R est la taille maximum de message de couche transport qui puisse être reçue (et réassemblée). TCP obtient MMS_R et MMS_S de la couche IP ; voir l'appel générique GET_MAXSIZES au paragraphe 3.4.

Discussion :
Le choix de la taille de segment TCP a un fort effet sur les performances. Les plus grands segments accroissent le débit en amortissant la redondance du traitement de taille d'en-tête et du nombre de datagramme sur plus d'octets de données ; cependant, si le paquet est si grand qu'il cause la fragmentation IP, l'efficacité chute considérablement si des fragments sont perdus [IP:9]. Certaines mises en œuvre TCP n'envoient une option MSS que si l'hôte de destination est sur un réseau non connecté. Cependant, en général la couche TCP peut n'avoir pas les informations appropriées pour prendre cette décision, aussi est il préférable de laisser à la couche IP la tâche de déterminer une MTU convenable pour le chemin Internet. On recommande donc que TCP envoie toujours l'option (si ce n'est pas 536) et que la couche IP détermine MMS_R comme spécifié en 3.3.3 et 3.4. Un mécanisme proposé de couche IP pour mesurer la MTU modifierait alors la couche IP sans changer TCP.

4.2.2.7 Somme de contrôle TCP : RFC-793 paragraphe 3.1

À la différence de la somme de contrôle UDP (voir au paragraphe 4.1.3.4), la somme de contrôle TCP n'est jamais facultative. L'envoyeur DOIT la générer et le receveur DOIT la vérifier.

4.2.2.8 Diagramme TCP d'état de connexion : RFC-793 paragraphe 3.2, page 23

Ces diagrammes posent plusieurs problèmes :
(a) La flèche qui va de SYN-SENT à SYN-RCVD devrait être marquée "snd SYN,ACK", pour être en accord avec le texte de la page 68 et la Figure 8.
(b) Il pourrait y avoir une flèche de l'état SYN-RCVD à l'état LISTEN, à condition de recevoir un RST après une ouverture passive (voir le texte page 70).
(c) Il est possible d'aller directement de l'état FIN-WAIT-1 à l'état TIME-WAIT (voir la page 75 de la spécification).

4.2.2.9 Choix du numéro de séquence initiale : RFC-793, paragraphe 3.3, page 27

TCP DOIT utiliser le choix spécifié piloté par l'horloge des numéros de séquence initiale.

4.2.2.10 Tentatives d'ouverture simultanées : RFC-793, paragraphe 3.4, page 32

Il y a une erreur à la Figure 8 : le paquet ligne 7 devrait être identique au paquet de la ligne 5.

TCP DOIT prendre en charge les tentatives d'ouverture simultanées.

Discussion :
Les développeurs de mises en œuvre sont parfois surpris si deux applications essayent simultanément de se connecter l'une à l'autre, une seule connexion est générée au lieu de deux. C'est une décision intentionnelle de conception ; n'essayez pas de le "réparer".

4.2.2.11 Récupération de l'ancien état SYN dupliqué : RFC-793, paragraphe 3.4, page 33

Noter qu'une mise en œuvre de TCP DOIT garder trace du fait qu'une connexion a atteint l'état SYN_RCVD comme résultat d'un OPEN passif ou d'un OPEN actif.

4.2.2.12 Segment RST : RFC-793, paragraphe 3.4

TCP DEVRAIT permettre qu'un segment RST reçu comporte des données.

Discussion :
Il a été suggéré qu'un segment RST puisse contenir du texte ASCII qui codait et expliquait la cause du RST. Aucune norme n'a encore été établie pour de telles données.

4.2.2.13 Fermeture d'une connexion : RFC-793, paragraphe 3.5

Une connexion TCP peut se terminer de deux façons : (1) la séquence de clôture normale de TCP en utilisant une prise de contact FIN, et (2) une "interruption" dans laquelle un ou plusieurs segments RST sont envoyés et où l'état de connexion est immédiatement éliminé. Si une connexion TCP est close par le site distant, l'application locale DOIT être informée de ce qu'elle est close normalement ou interrompue.

La séquence de clôture TCP normale délivre les données de mémoire tampon de façon fiable dans les deux directions. Comme les deux directions d'une connexion TCP sont closes de façon indépendante, il est possible qu'une connexion soit "à demi close," c'est-à-dire, close dans une seule direction, et il est permis à un hôte de continuer d'envoyer des données dans la direction ouverte sur une connexion à demi close.

Un hôte PEUT mettre en œuvre une séquence de clôture TCP "en semi duplex", de sorte qu'une application qui a invoqué CLOSE ne peut pas continuer à lire des données à partir de la connexion. Si un tel hôte produit un appel CLOSE alors que des données reçues sont toujours en cours dans TCP, ou si de nouvelles données sont reçues après l'invocation de CLOSE, TCP DEVRAIT envoyer un RST pour montrer que des données ont été perdues.

Lorsque une connexion est close de façon active, elle DOIT rester dans l'état TIME-WAIT pendant une durée de 2xMSL (Maximum Segment Lifetime, *durée de vie maximum de segment*). Cependant, elle PEUT accepter qu'un nouveau SYN provenant du TCP distant rouvre la connexion directement à partir de l'état TIME-WAIT, si :
(1) elle réalloue son numéro de séquence initial pour la nouvelle connexion supérieur au plus grand numéro de séquence utilisé sur la précédente incarnation de la connexion, et
(2) elle retourne à l'état TIME-WAIT si le SYN se révèle être un vieux duplicata.

Discussion :
La clôture en bilatéral préservant les données de TCP est une caractéristique qui n'est pas incluse dans le protocole de transport ISO TP4 analogue.

Certains systèmes n'ont pas mis en œuvre la connexion semi close, vraisemblablement parce qu'elle ne rentre pas dans le modèle ouvert/fermé de leurs systèmes d'exploitation particuliers. Sur ces systèmes, une fois qu'une application a invoqué CLOSE, elle ne peut plus lire des entrées de données provenant de la connexion ; c'est ce qu'on appelle une séquence de clôture TCP "semi duplex".

L'algorithme de clôture en douceur de TCP exige que l'état de la connexion reste défini (au moins) à une des extrémités de la connexion, pour une période de temporisation de 2xMSL, c'est-à-dire 4 minutes. Durant cette période, la paire (prise distante, prise locale) qui définit la connexion est occupée et ne peut être réutilisée. Pour abréger la durée d'occupation d'une paire d'accès donnée, certains TCP permettent qu'un nouveau SYN soit accepté dans l'état TIME-WAIT.

4.2.2.14 Communication de données : RFC-793, paragraphe 3.7, page 40

Depuis la rédaction de la RFC-793, il y a eu un travail intensif sur les algorithmes TCP pour réaliser des communications de données efficaces. Les paragraphes ultérieurs du présent document décrivent les algorithmes exigés et recommandés pour que TCP détermine quand envoyer les données (paragraphe 4.2.3.4), quand envoyer un accusé de réception (paragraphe 4.2.3.2), et quand mettre à jour la fenêtre (paragraphe 4.2.3.3).

Discussion :
Une importante question pour les performances est celle du "syndrome de la fenêtre folle" (SWS, *Silly Window Syndrome*) [RFC0813], un schéma stable de petits mouvements croissants de fenêtre résultant en performances TXP extrêmement faibles. Les algorithmes pour éviter le SWS sont décrits ci-dessous pour le côté émetteur (paragraphe 4.2.3.4) et pour le côté récepteur (paragraphe 4.2.3.3).

En bref, SWS est causé par l'avancée du bord droit de la fenêtre par le receveur chaque fois qu'il a un nouvel espace de mémoire tampon disponible pour recevoir des données et par l'utilisation de la part de l'émetteur de toute fenêtre supplémentaire, si petite soit elle, pour envoyer plus de données [RFC0813]. Le résultat peut être un schéma stable d'envoi de minuscules segments de données, même si l'émetteur et le receveur ont tous deux un grand espace de mémoire tampon total pour la connexion. Le SWS ne peut survenir que durant la transmission de grandes quantités de données ; si la connexion est moins sollicitée, le problème va disparaître. Il est causé par la mise en œuvre directe normale de la gestion de fenêtre, mais les algorithmes d'émission et de réceptions donnés ci-dessous l'évitent.

Une autre importante question de performance de TCP est que certaines applications, en particulier de connexion à distance sur des hôtes qui n'acceptent qu'un caractère à la fois, tendent à envoyer des flux de segments de données de un octet. Pour éviter les impasses, chaque appel TCP SEND provenant de telles applications doit être "poussé", soit explicitement par l'application, soit alors implicitement par TCP. Le résultat peut être un flux de segments TCP contenant chacun un octet de données, ce qui fait une utilisation très inefficace de l'Internet et contribue à son encombrement. L'algorithme de Nagle décrit au paragraphe 4.2.3.4 fournit une solution simple et efficace à ce problème. Il a pour effet de grouper les caractères sur les connexions Telnet ; cela peut surprendre au début les utilisateurs habitués à un écho d'un seul caractère, mais l'avis de l'utilisateur n'est pas demandé. Noter que l'algorithme de Nagle et l'algorithme d'évitement du SWS jouent un rôle complémentaire dans l'amélioration des performances. L'algorithme de Nagle décourage l'envoi de petits segments lorsque l'accroissement des données à envoyer se fait par petites quantités, alors que l'algorithme d'évitement de SWS décourage les petits segments résultants de l'avancement du bord droit de la fenêtre par petits incréments.

Une mise en œuvre négligente peut envoyer deux segments d'accusé de réception ou plus par segment de données reçu. Par exemple, supposons que le receveur accuse immédiatement réception de chaque segment de données. Lorsque le programme d'application consomme ensuite les données et augmente à nouveau l'espace de mémoire tampon disponible de réception, le receveur peut envoyer un second segment d'accusé de réception pour mettre à jour la fenêtre chez l'émetteur. Le cas extrême survient avec des segments d'un seul caractère sur les connexions TCP utilisant le protocole Telnet pour le service de connexion à distance. Il a été observé que certaines mises en œuvre génèrent trois segments en retour pour

chaque segment de un caractère entrant : respectivement (1) l'accusé de réception, (2) une augmentation d'un octet de la fenêtre, et (3) le caractère d'écho.

4.2.2.15 Fin de temporisation de retransmission : RFC-793 paragraphe 3.7, page 41

L'algorithme suggéré dans la RFC-793 pour le calcul de la temporisation de retransmission n'a pas de vice connu ; voir au paragraphe 4.2.3.1 ci-dessous.

Un travail récent de Jacobson [TCP:7] sur l'encombrement Internet et la stabilité de retransmission TCP a produit un algorithme de transmission qui combine "démarrage lent" *(slow start)* avec "évitement d'encombrement" *(congestion avoidance)*. TCP DOIT mettre en œuvre cet algorithme.

Si un paquet retransmis est identique au paquet original (ce qui n'implique pas seulement que les frontières des données n'aient pas changé, mais aussi que les champs de fenêtre et d'accusé de réception de l'en-tête n'aient pas changé) le même champ Identification IP PEUT alors être utilisé (voir au paragraphe 3.2.1.5).

Mise en œuvre :
Certains développeurs de mises en œuvre de TCP ont choisi de "mettre en paquet" le flux de données, c'est-à-dire, de collecter les frontières de segment lorsque les segments sont envoyés pour la première fois et de mettre ces segments dans une "file d'attente de retransmission" jusqu'à ce qu'il en ait été accusé réception. Une autre conception (peut être plus simple) est de différer la mise en paquet jusqu'à ce que chaque donnée d'heure soit transmise ou retransmise, de sorte qu'il n'y a plus de file d'attente de retransmission de segment.

Dans une mise en œuvre avec file d'attente de retransmission de segment, les performance de TCP peuvent être améliorées par la remise en paquet des segments qui attendent qu'il en soit accusé réception lorsque survient la fin de temporisation de la première retransmission. C'est-à-dire, les segments en suspens qui conviennent seront combinés en un segment de la taille maximale, avec une nouvelle valeur Identification IP. TCP retiendra alors ce segment combiné dans la file d'attente de retransmission jusqu'à ce qu'il en soit accusé réception. Cependant, si les deux premiers segments dans la file d'attente de retransmission font au total plus qu'un segment de la taille maximum, TCP ne retransmettra que le premier segment en utilisant le champ Identification IP original.

4.2.2.16 Gestion de la fenêtre : RFC-793, paragraphe 3.7, page 41

Un receveur TCP NE DEVRAIT PAS réduire la fenêtre, c'est-à-dire, déplacer le bord droit de la fenêtre vers la gauche. Cependant, un TCP émetteur DOIT être robuste à l'égard de la réduction de fenêtre, ce qui peut entraîner le passage en négatif de "la fenêtre utilisable" (voir au paragraphe 4.2.3.4).

Si cela arrive; l'émetteur NE DEVRAIT PAS envoyer de nouvelles données, mais DEVRAIT retransmettre normalement les vieilles données dont il n'a pas été accusé réception entre SND.UNA et SND.UNA+SND.WND. L'émetteur PEUT aussi retransmettre des vieilles données au-delà de SND.UNA+SND.WND, mais NE DEVRAIT PAS faire expirer le temps de la connexion s'il n'a pas été accusé réception de données au-delà du bord droit de la fenêtre. Si la fenêtre se réduit à zéro, TCP DOIT la vérifier de la façon standard (voir le paragraphe suivant).

Discussion :
De nombreuses mises en œuvre de TCP sont troublées si la fenêtre se réduit à partir de la droite après que les données aient été envoyées dans une fenêtre plus grande. Noter que TCP a une heuristique pour choisir la dernière mise à jour de fenêtre en dépit d'une réorganisation possible des datagrammes ; il en résulte qu'il peut ignorer une mise à jour de fenêtre avec une fenêtre plus petite que celle offerte précédemment si ni le numéro de séquence ni le numéro d'accusé de réception n'ont augmenté.

4.2.2.17 Vérification des fenêtres zéro : RFC-793, paragraphe 3.7, page 42

La vérification (offerte) des fenêtres zéro DOIT être pris en charge.
TCP PEUT garder close indéfiniment sa fenêtre de réception offerte. Tant que le TCP de réception continue d'envoyer des accusés de réception en réponse aux segments de vérification, le TCP émetteur DOIT permettre que la connexion reste ouverte.

Discussion : Il est extrêmement important de se souvenir que les segments ACK (accusé de réception) qui ne contiennent pas de données ne sont pas transmis de façon fiable par TCP. Si la vérification des fenêtres zéro n'est pas prise en charge, une connexion peut rester en instance à tout jamais si un segment ACK qui rouvre la fenêtre est perdu.

Le délai d'ouverture d'une fenêtre zéro survient généralement lorsque l'application receveuse arrête de prendre des données provenant de son TCP. Par exemple, considérons une application d'imprimante automatique, stoppée parce que l'imprimante est à court de papier.

L'hôte émetteur DEVRAIT envoyer la première vérification de fenêtre zéro lorsqu'une fenêtre zéro a existé pendant la période de temporisation de la retransmission (voir au paragraphe 4.2.2.15) et DEVRAIT augmenter exponentiellement l'intervalle entre les vérifications successives.

Discussion :
Cette procédure minimise le délai si la condition de fenêtre zéro est due à la perte d'un segment ACK contenant une mise à jour d'ouverture de fenêtre. Une augmentation exponentielle de temporisation est recommandée, éventuellement avec un intervalle maximum qui n'est pas spécifié ici. Cette procédure est similaire à celle de l'algorithme de retransmission, et il peut être possible de combiner les deux procédures dans la mise en œuvre.

4.2.2.18 Appels OPEN passifs : RFC-793, paragraphe 3.8

Tout appel OPEN passif crée un nouvel enregistrement de connexion en état LISTEN, ou retourne une erreur ; il NE DOIT PAS affecter d'enregistrement de connexion précédemment créé.

Un TCP qui accepte plusieurs utilisateurs concurrents DOIT fournir une commande OPEN qui permette fonctionnellement à une application d'être en état LISTEN sur un accès alors qu'un bloc de connexion avec le même accès local est dans l'état SYN-SENT ou SYN-RECEIVED.

Discussion :
Certaines applications (par exemple, les serveurs SMTP) peuvent avoir besoin de traiter plusieurs tentatives de connexion presque simultanément. La probabilité d'échec d'une tentative de connexion est réduite en donnant à l'application des moyens pour écouter une nouvelle connexion pendant le temps où une tentative de connexion antérieure passe par la prise de contact ternaire.

Mise en œuvre :
Les mises en œuvre qui acceptent des ouvertures concurrentes peuvent permettre plusieurs appels OPEN passif, ou peuvent permettre le "clonage" de connexions en état LISTEN à partir d'un seul appel OPEN passif.

4.2.2.19 Durée de vie : RFC-793, paragraphe 3.9, page 52

La RFC-793 spécifiait que TCP devait demander à la couche IP d'envoyer des segments TCP avec TTL = 60. C'est obsolète ; la valeur de TTL utilisée pour envoyer des segments TCP DOIT être configurable. Voir l'exposé au paragraphe 3.2.1.7.

4.2.2.20 Traitement d'événement : RFC-793, paragraphe 3.9

Bien que ce ne soit pas strictement exigé, un TCP DEVRAIT être capable de mettre en file d'attente des segments TCP décalés. Il faut changer le "peut" de la dernière phrase du premier paragraphe de la page 70 en "devrait".

Discussion :
Certaines mises en œuvre de petits hôtes ont omis la mise en file d'attente des segments à cause d'un espace limité de mémoire tampon. On peut s'attendre à ce que cette omission ait un effet néfaste sur le débit de TCP, car la perte d'un seul segment cause l'apparition de tous les segments ultérieurs comme "hors séquence".

En général, le traitement des segments reçus DOIT être mis en œuvre pour agréger les segments ACK chaque fois que possible. Par exemple, si le TCP traite une série de segments mis en file d'attente, il DOIT les traiter tous avant d'envoyer aucun segment ACK.

Voici quelques corrections et notes sur la correction d'erreur détaillées sur le paragraphe de traitement d'événement de la RFC-793.

(a) Appel CLOSE, état CLOSE-WAIT, p. 61 : entrer dans l'état LAST-ACK, et non pas CLOSING.

(b) État LISTEN, vérifier SYN (pages 65, 66) : avec un bit SYN, si le compartiment sécurité ou la préséance est faux pour le segment, un rétablissement est envoyé. La mauvaise forme de rétablissement est indiquée dans le texte ; elle devrait être :

<SEQ=0><ACK=SEG.SEQ+SEG.LEN><CTL=RST,ACK>

(c) État SYN-SENT, vérifier SYN, p. 68 : Lorsque la connexion entre dans l'état ESTABLISHED, les variables suivantes doivent être établies :
SND.WND <- SEG.WND
SND.WL1 <- SEG.SEQ
SND.WL2 <- SEG.ACK

(d) Vérifier la sécurité et la préséance, p. 71 : le premier en-tête "ESTABLISHED STATE" devrait réellement être une liste de tous les états autres que SYN-RECEIVED : ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, et TIME-WAIT.

(e) Vérifier le bit SYN, p. 71 : "Dans l'état SYN-RECEIVED et si la connexion a été initialisée avec un OPEN passif, repasser alors cette connexion à l'état LISTEN et retourner. Autrement...".

(f) Vérifier le champ ACK, état SYN-RECEIVED, p. 72 : Lorsque la connexion entre dans l'état ESTABLISHED, les variables dont la liste figure en (c) doivent être établies.

(g) Vérifier le champ ACK, état ESTABLISHED, p. 72 : Le ACK est un duplicata si SEG.ACK ≤ SND.UNA (le = était omis). De même, la fenêtre devrait être mise à jour si : SND.UNA ≤ SEG.ACK ≤ SND.NXT.

(h) USER TIMEOUT, p. 77 : Il serait mieux de notifier à l'application la fin de temporisation plutôt que de laisser TCP forcer la clôture de la connexion. Cependant, voir aussi le paragraphe 4.2.3.5.

4.2.2.21 Accusé de réception des segments en file d'attente : RFC-793, paragraphe 3.9

TCP PEUT envoyer un segment ACK accusant réception de RCV.NXT lorsqu'un segment valide arrive et qui est dans la fenêtre mais pas sur le bord gauche de la fenêtre.

Discussion :
La RFC-793 (voir page 74) était ambiguë sur la question de savoir si un segment ACK devrait ou non être envoyé lors de la réception d'un segment décalé, c'est-à-dire, lorsque SEG.SEQ n'est pas égal à RCV.NXT.

Une raison de l'envoi d'accusé de réception des segments décalés pourrait être de prendre en charge un algorithme expérimental connu sous le nom de "retransmission rapide". Avec cet algorithme, l'émetteur utilise le ACK "redondant" pour en déduire qu'un segment a été perdu avant l'expiration du temporisateur de retransmission. Il compte le nombre de fois qu'un ACK a été reçu avec la même valeur de SEG.ACK et avec le même bord droit de fenêtre. Si plus d'un nombre seuil de tels ACK est reçu, les segments contenant les octets commençant à SEG.ACK sont supposés avoir été perdus et sont retransmis, sans attendre une fin de temporisation. Le seuil est choisi pour compenser la réorganisation vraisemblable maximum de segments dans l'Internet. Il n'y a pas encore assez d'expérience de l'algorithme de retransmission rapide pour déterminer quelle est son utilité.

4.2.3 Questions spécifiques

4.2.3.1 Calcul de la temporisation de retransmission

Un hôte TCP DOIT mettre en œuvre l'algorithme de Karn et l'algorithme de Jacobson pour le calcul de la fin de temporisation de retransmission ("RTO").

- L'algorithme de Jacobson pour le calcul du délai d'aller retour ("RTT") lissé incorpore une simple mesure de la variance [TCP:7].

- L'algorithme de Karn pour le choix des mesure de RTT garantit que des délais d'aller retour ambigus ne vont pas fausser le calcul du délai lissé d'aller retour [TCP:6].

Cette mise en œuvre DOIT aussi inclure la "temporisation exponentielle" pour des valeurs successives de RTO pour le même segment. La retransmission des segments SYN DEVRAIT utiliser le même algorithme que pour les segments de données.

Discussion :
Deux problèmes sont rencontrés avec les calculs de RTO spécifiés dans la RFC-793. D'abord, la mesure précise des RTT est difficile lorsqu'il y a des retransmissions. Ensuite, l'algorithme pour calculer le délai lissé d'aller retour est inadéquat [TCP:7], parce qu'il est incorrectement supposé que la variance des valeurs de RTT devrait être faible et constante. Ces problèmes ont été résolus respectivement par les algorithmes de Karn et de Jacobson.

L'accroissement des performances résultant de l'utilisation de ces améliorations varie dans des proportions considérables. L'algorithme de Jacobson pour incorporer la variance du RTT mesuré est particulièrement important sur une liaison à basse vitesse, où la variation naturelle de la taille de paquet cause une grande variation du RTT. Un fabricant a trouvé que l'utilisation de la liaison sur une ligne à 9,6 kbit va de 10 à 90 % par suite de la mise en œuvre de l'algorithme de variance de Jacobson sur TCP.

Les valeurs suivantes DEVRAIENT être utilisées pour initialiser les paramètres d'estimation pour une nouvelle connexion :
(a) RTT = 0 seconde.
(b) RTO = 3 secondes. (La variance lissée est à initialiser à la valeur qui va résulter de ce RTO).

Les valeurs recommandées de limite supérieure et inférieure du RTO sont inadéquates sur de grands internets. La limite inférieure DEVRAIT être mesurée en fractions de seconde (pour s'accommoder des LAN à haut débit) et la limite supérieure devrait être 2*MSL, c'est-à-dire, 240 secondes.

Discussion :
L'expérience a montré que ces valeurs d'initialisation sont raisonnables, et que dans tous les cas les algorithmes de Karn et Jacobson rendent le comportement de TCP raisonnablement insensible au choix des paramètres initiaux.

4.2.3.2 Quand envoyer un segment ACK

Un hôte qui reçoit un flux de segments de données TCP peut accroître l'efficacité à la fois dans l'Internet et chez les hôtes en voyant moins d'un segment ACK (accusé de réception) par segment de données reçu ; c'est connu sous le nom de "ACK retardé" [RFC0813].

TCP DEVRAIT mettre en œuvre un ACK retardé, mais un ACK ne devrait pas être excessivement retardé ; en particulier, le retard DOIT être inférieur à 0,5 seconde, et dans un flux de segments de taille complète, il DEVRAIT y avoir un ACK pour au moins un segment par seconde.

Discussion :
Un ACK retardé donne à l'application l'opportunité de mettre à jour la fenêtre et peut-être d'envoyer une réponse immédiate. En particulier, dans le cas de connexion distante en mode caractères, un ACK retardé peut réduire le nombre de segments envoyés par le serveur d'un facteur de 3 (ACK, mise à jour de fenêtre, et caractère d'écho tous combinés en un seul segment).

De plus, sur certains grands hôtes multi-utilisateurs, un ACK retardé peut substantiellement réduire la redondance de traitement de protocole en réduisant le nombre total de paquets à traiter [RFC0813]. Cependant, des retards excessifs sur les ACK peuvent perturber le délai d'aller-retour et les algorithmes de "synchronisation " de paquets [TCP:7].

4.2.3.3 Quand envoyer une mise à jour de fenêtre

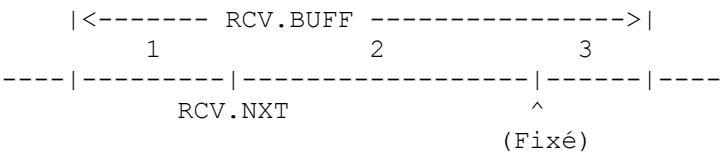
TCP DOIT inclure un algorithme d'évitement de SWS chez le receveur [RFC0813].

Mise en œuvre :
L'algorithme d'évitement de SWS du receveur détermine quand le bord droit de fenêtre peut être avancé ; c'est ce qu'on appelle traditionnellement "mettre à jour la fenêtre". Cet algorithme se combine avec l'algorithme d'ACK retardé (voir au paragraphe 4.2.3.2) pour déterminer quand un segment ACK contenant la fenêtre actuelle va réellement être envoyé au receveur. On utilise la notation de la RFC-793 ; voir les Figures 4 et 5 de ce document.

La solution à la SWS de réception est d'éviter d'avancer le bord droit de la fenêtre RCV.NXT+RCV.WND en petits incréments, même si les données sont reçues du réseau en petits segments.

Supposons que l'espace total de mémoire tampon de réception soit de RCV.BUFF. À tout instant donné, RCV.USER octets de ce total peuvent être liés à des données qui ont été reçues et dont il a été accusé réception mais que le processus d'utilisateur n'a pas encore consommées. Lorsque la connexion est au repos, RCV.WND = RCV.BUFF et RCV.USER = 0.

Garder le bord droit de la fenêtre fixe lorsque les données arrivent et qu'il en est accusé réception exige que le receveur offre moins que son espace complet de mémoire tampon, c'est-à-dire que le receveur doit spécifier un RCV.WND qui garde RCV.NXT+RCV.WND constant lorsque RCV.NXT s'accroît. Et donc, l'espace total de mémoire tampon RCV.BUFF est généralement divisé en trois parties :



1 - RCV.USER = données reçues mais pas encore consommées ;
2 - RCV.WND = espace communiqué à l'émetteur ;
3 - Réduction = espace disponible mais pas encore communiqué.

L'algorithme d'évitement de SWS suggéré pour le receveur est de garder RCV.NXT+RCV.WND fixé jusqu'à ce que la réduction satisfasse :

$$\text{RCV.BUFF} - \text{RCV.USER} - \text{RCV.WND} \geq \min(\text{Fr} * \text{RCV.BUFF}, \text{Eff.snd.MSS})$$

où Fr est une fraction dont la valeur recommandée est 1/2, et Eff.snd.MSS est la MSS effective d'envoi pour la connexion (voir au paragraphe 4.2.2.6). Lorsque l'inégalité est satisfaite, RCV.WND est réglé à RCV.BUFF-RCV.USER.

Noter que l'effet général de cet algorithme est d'avancer RCV.WND en incréments de Eff.snd.MSS (pour des mémoires tampon de réception réalistes : Eff.snd.MSS < RCV.BUFF/2). Noter aussi que le receveur doit utiliser son propre Eff.snd.MSS, en supposant qu'il est le même que celui de l'émetteur.

4.2.3.4 Quand envoyer les données

TCP DOIT inclure un algorithme d'évitement de SWS chez l'émetteur.

TCP DEVRAIT mettre en œuvre l'algorithme de Nagle [RFC0896] pour regrouper les segments courts. Cependant, il DOIT y avoir un moyen pour qu'une application désactive l'algorithme de Nagle sur une connexion individuelle. Dans tous les cas, l'envoi des données est aussi soumis aux limitations imposées par l'algorithme de démarrage lent (paragraphe 4.2.2.15).

Discussion :
L'algorithme de Nagle est généralement comme suit :

Si il y a des données dont il n'a pas été accusé réception (c'est-à-dire, SND.NXT > SND.UNA), le TCP émetteur met en mémoire tampon toutes les données d'utilisateur (sans considération du bit PSH), jusqu'à ce qu'il ait été accusé réception des données en cours ou que TCP puisse envoyer un segment de taille complète (Eff.snd.MSS octets ; voir au paragraphe 4.2.2.6).

Certaines applications (par exemple, la mise à jour de fenêtre d'affichage en temps réel) exigent que l'algorithme de Nagle soit désactivé, de façon que les petits segments de données puissent être traités en direct au débit maximum.

Mise en œuvre :
L'algorithme d'évitement de SWS de l'émetteur est plus difficile que celui du receveur, parce que l'émetteur ne connaît pas (directement) l'espace total RCV.BUFF de mémoire tampon du receveur. Il a été trouvé une approche qui fonctionne bien et qui consiste pour l'émetteur à calculer Max(SND.WND), la fenêtre d'envoi maximum qui s'est rencontrée jusqu'à présent sur la connexion, et d'utiliser cette valeur comme estimation de RCV.BUFF. Malheureusement, cela n'est qu'une estimation ; le receveur peut à tout moment réduire la taille de RCV.BUFF. Pour éviter de tomber dans une impasse, il est nécessaire d'avoir une temporisation pour forcer la transmission des données, prenant le pas sur l'algorithme d'évitement de SWS. En pratique, cette temporisation devrait rarement survenir.

La "fenêtre utilisable" [RFC0813] est :

U = SND.UNA + SND.WND - SND.NXT

c'est-à-dire, la fenêtre offerte moins la quantité de données envoyées mais dont il n'a pas été accusé réception. Si D est la quantité de données en file d'attente dans le TCP d'envoi mais non encore envoyées, l'ensemble des règles suivantes est alors recommandé.

Données envoyées :
(1) si un segment de la taille maximum peut être envoyé, c'est-à-dire, si : min(D,U) ≥ Eff.snd.MSS ;
(2) ou si les données sont poussées et que toutes les données en file d'attente peuvent maintenant être envoyées, c'est-à-dire, si : [SND.NXT = SND.UNA et] PUSHED et D ≤ U (la condition entre crochets est imposée par l'algorithme de Nagle) ;
(3) ou si au moins une fraction Fs de la fenêtre maximum peut être envoyée, c'est-à-dire, si : [SND.NXT = SND.UNA et] min(D.U) ≥ Fs * Max(SND.WND);
(4) ou si les données sont poussées et que survient la fin de temporisation d'outrepassement.

Ici Fs est une fraction dont la valeur recommandée est 1/2. La temporisation d'outrepassement devrait être dans la gamme de 0,1 – 1,0 seconde. Il peut être pratique de combiner ce temporisateur avec celui utilisé pour la vérification des fenêtres zéro (paragraphe 4.2.2.17).

Finalement, noter que l'algorithme d'évitement de SWS qui vient d'être spécifié est à utiliser à la place de l'algorithme côté émetteur contenu dans [RFC0813].

4.2.3.5 Défaillances de connexion TCP

Une retransmission excessive du même segment par TCP indique une défaillance de l'hôte distant ou du chemin Internet. Cette défaillance peut être de longue ou courte durée. La procédure suivante DOIT être utilisée pour traiter les retransmissions excessives de segments de données [RFC0816]:

(a) Il y a deux seuils R1 et R2 qui mesurent la quantité de retransmissions qui sont survenues pour le même segment. R1 et R2 peuvent être mesurés en unités de temps ou comme compte de retransmissions.

(b) Lorsque le nombre de transmissions du même segment atteint ou dépasse le seuil R1, on passe un avis négatif (voir au paragraphe 3.3.1.4) à la couche IP, pour déclencher un diagnostic de routeur mort.

(c) Lorsque le nombre de transmissions du même segment atteint le seuil R2 supérieur à R1, clore la connexion.

(d) Une application DOIT être capable de régler la valeur de R2 pour une connexion particulière. Par exemple, une application interactive pourrait régler R2 à "infini," donnant à l'utilisateur le contrôle sur le moment de la déconnexion.

(e) TCP DEVRAIT informer l'application du problème de livraison (sauf si de telles informations ont été désactivées par l'application ; voir au paragraphe 4.2.4.1), lorsque R1 est atteint et avant R2. Cela permettra à un programme d'application de connexion à distance (User Telnet) d'informer l'utilisateur, par exemple.

La valeur de R1 DEVRAIT correspondre au moins à trois retransmissions, au RTO en cours. La valeur de R2 DEVRAIT correspondre au moins à 100 secondes.

Une tentative d'ouvrir une connexion TCP pourrait échouer avec d'excessives retransmissions du segment SYN ou par la réception d'un segment RST ou d'un Accès injoignable ICMP. Les retransmissions de SYN DOIVENT être traitées de la façon générale qui vient d'être décrite pour les retransmissions de données, y compris la notification à la couche d'application.

Cependant, les valeurs de R1 et R2 peuvent être différentes pour SYN et les segments de données. En particulier, R2 pour un segment SYN DOIT être réglé assez grand pour fournir la retransmission de segments pendant au moins 3 minutes. Bien sûr, l'application peut clore la connexion (c'est-à-dire, abandonner la tentative d'ouverture) plus tôt.

Discussion : Certains chemins Internet ont des temps d'établissement significatifs, et le nombre de ces chemins augmentera vraisemblablement à l'avenir.

4.2.3.6 Garder en vie TCP

Les développeurs PEUVENT inclure des "garder-en-vie" *(Keep-Alive)* dans leurs mises en œuvre TCP, bien que cette pratique ne soit pas universellement acceptée. Si les garder-en-vie sont inclus, l'application DOIT être capable de les activer ou les désactiver pour chaque connexion TCP, et ils DOIVENT être désactivés par défaut.

Les paquets garder-en-vie ne DOIVENT être envoyés que lorsque aucun paquet de données ou d'accusé de réception n'a été reçu pour la connexion dans un certain intervalle. Cet intervalle DOIT être configurable et DOIT par défaut n'être pas inférieur à deux heures.

Il est extrêmement important de se souvenir que les segments ACK qui ne contiennent pas de données ne sont pas transmis de façon fiable par TCP. Par conséquent, si un mécanisme de maintien en vie est mis en œuvre, il NE DOIT PAS interpréter le défaut de réponse à toute vérification spécifique comme une connexion morte.

Une mise en œuvre DEVRAIT envoyer un segment garder-en-vie sans données ; cependant, il PEUT être configurable d'envoyer un segment garder-en-vie contenant un octet poubelle, pour la compatibilité avec les mises en œuvre TCP erronées.

Discussion :
Un mécanisme "garder-en-vie" teste périodiquement l'autre extrémité d'une connexion lorsque la connexion est par ailleurs inactive, même lorsqu'il n'y a pas de données à envoyer. La spécification TCP ne comporte pas de mécanisme garder-en-vie parce que cela peut : (1) causer l'interruption de connexions parfaitement aptes durant des défaillances Internet temporaires ; (2) consommer sans nécessité de la bande passante ("si personne n'utilise la connexion, qui se soucie de savoir si elle est encore bonne ?"); et (3) coûte de l'argent pour un chemin Internet qui facture les paquets.

Certaines mises en œuvre TCP, ont cependant inclus une mécanisme garder-en-vie. Pour confirmer qu'une connexion au repos est toujours active, ces mises en œuvre envoient un segment de vérification conçu pour provoquer une réponse de la part du TCP homologue. Un tel segment contient généralement SEG.SEQ = SND.NXT-1 et peut contenir ou non un octet de données poubelles. Noter que sur une connexion au repos SND.NXT = RCV.NXT, de sorte que ce SEG.SEQ sera en-dehors de la fenêtre. Donc, la vérification amène le receveur à retourner un segment d'accusé de réception, confirmant que la connexion est toujours en vie. Si l'homologue a abandonné la connexion du fait d'une partition du réseau ou d'un accident, il va répondre par un RST à la place du segment d'accusé de réception.

Malheureusement, certaines mises en œuvre TCP mal formées ne réussissent pas à répondre à un segment par SEG.SEQ = SND.NXT-1 si le segment ne contient pas de données. Autrement, une mise en œuvre pourrait déterminer si un homologue a répondu correctement aux paquets garder-en-vie sans octet de données poubelles.

Un mécanisme garder-en-vie TCP ne pourrait être invoqué que dans des applications de serveur qui autrement resteraient activées indéfiniment et consommeraient sans nécessité des ressources si un client connaît une défaillance ou interrompt une connexion durant une défaillance du réseau.

4.2.3.7 Rattachement multiple TCP

Si une application sur un hôte à rattachements multiples ne spécifie pas l'adresse IP locale lors d'une ouverture active d'une connexion TCP, le TCP DOIT alors demander à la couche IP de choisir une adresse IP locale avant d'envoyer le (premier) SYN. Voir la fonction GET_SRCADDR() au paragraphe 3.4.

Toutes les autres fois, un segment précédent a été envoyé ou reçu sur cette connexion, et TCP DOIT utiliser la même adresse locale que celle utilisée dans ces segments précédents.

4.2.3.8 Options IP

Lorsque les options reçues sont passées à TCP de la couche IP, TCP DOIT ignorer les options qu'il ne comprend pas.

TCP PEUT prendre en charge les options Horodatage et Route enregistrée.

Une application DOIT être capable de spécifier une route de source lorsqu'elle ouvre activement une connexion TCP, et celle-ci DOIT prendre le pas sur une route de source reçue dans un datagramme.

Lorsque une connexion TCP est OUVERTE passivement et qu'un paquet arrive avec une option Route de source IP complétée (contenant une route de retour), TCP DOIT sauvegarder la route de retour et l'utiliser pour tous les segments envoyés sur cette connexion. Si une route de source différente arrive dans un segment ultérieur, la dernière définition DEVRAIT subroger la précédente.

4.2.3.9 Messages ICMP

TCP DOIT agir sur un message d'erreur ICMP passé de la couche IP, et le diriger sur la connexion qui a créé l'erreur. Les informations nécessaires au démultiplexage peuvent être trouvées dans l'en-tête IP contenu au sein du message ICMP.

- Source éteinte
  TCP DOIT réagir à une Source éteinte en ralentissant la transmission sur la connexion. La procédure RECOMMANDÉE est pour une Source éteinte de déclencher un "démarrage lent," comme si une fin de temporisation de retransmission était intervenue.

- Destination injoignable -- codes 0, 1, 5
  Comme ces messages Injoignable indiquent des conditions d'erreur douce, TCP NE DOIT PAS interrompre la connexion, et DEVRAIT rendre les informations disponibles à l'application.

Discussion :
TCP pourrait rapporter la condition d'erreur douce directement à la couche d'application avec un appel à la routine ERROR_REPORT, ou il pourrait simplement noter le message et ne le rapporter à l'application que quand et si la connexion TCP arrive en fin de temporisation.

- Destination injoignable -- codes 2-4
  Ces sont des conditions d'erreur dure, aussi TCP DEVRAIT interrompre la connexion.

- Délai dépassé -- codes 0, 1
  Ceci devrait être traité de la même façon que les codes 0, 1, 5 Destination injoignable (voir ci-dessus).

- Problème de paramètre
  Ceci devrait être traité de la même façon que les codes 0, 1, 5 Destination injoignable 5 (voir ci-dessus).

4.2.3.10 Validation d'adresse distante

Une mise en œuvre de TCP DOIT rejeter comme erreur un appel local OPEN pour une adresse IP distante invalide (par exemple, une adresse de diffusion ou de diffusion groupée).

Un SYN entrant avec une adresse de source invalide doit être ignoré par TCP ou par la couche IP (voir au paragraphe 3.2.1.3).

Une mise en œuvre de TCP DOIT éliminer en silence un segment SYN entrant qui est adressé à une adresse de diffusion ou de diffusion groupée.

4.2.3.11 Schémas de trafic TCP

Mise en œuvre :
La spécification du protocole TCP [RFC0793] donne beaucoup de liberté à la mise en œuvre pour concevoir les algorithmes qui commandent le flux de messages sur la connexion – la mise en paquets, la gestion de fenêtre, l'envoi des accusés de réception, etc. Ces décisions de conception sont difficiles parce que TCP doit s'adapter à une large gamme de schémas de trafic. L'expérience a montré qu'un développeur de TCP doit vérifier la conception sur deux schémas de trafic extrêmes :

- Segments d'un seul caractère
  Même si l'émetteur utilise l'algorithme de Nagle, lorsque une connexion TCP porte du trafic de connexion distante à travers un LAN à faible retard, le receveur va généralement obtenir un flux de segments d'un seul caractère. Si le mode d'écho du terminal distant est activé, le système du receveur va généralement faire écho à chaque caractère quand il est reçu.

- Transfert en vrac

Lorsque TCP est utilisé pour du transfert en vrac, le flux de données devrait être (presque) entièrement constitué de segments de la taille du MSS effectif.
Bien que TCP utilise un espace de numéros de séquence d'une granularité d'un octet, dans le mode de transfert en vrac cette opération devrait être comme si TCP utilisait un espace de séquence qui ne compte que les segments.

L'expérience a montré de plus qu'un seul TCP peut effectivement et efficacement traiter ces deux extrêmes.

L'outil le plus important pour vérifier une nouvelle mise en œuvre de TCP est un programme de suivi des paquets. Il y a beaucoup d'expériences qui montrent l'importance du suivi de divers schémas de trafic avec d'autres mises en œuvre de TCP et de l'étude attentive des résultats.

4.2.3.12 Efficacité

Mise en œuvre :
Une large expérience conduit aux suggestions suivantes pour une mise en œuvre efficace de TCP :

(a) Ne pas copier les données
Dans le transfert de données en vrac, les principales tâches intensives du CPU sont de copier les données d'un registre dans l'autre et de faire des sommes de contrôle sur les données. Il est vital de minimiser le nombre de copies des données TCP. Comme la limitation ultime de vitesse peut être d'installer les données à travers le bus mémoire, il peut être utile de combiner la copie avec la somme de contrôle, en effectuant les deux avec une seule extraction de mémoire.

(b) Effectuer à la main le sous-programme de somme de contrôle
Un bon sous-programme de somme de contrôle TCP est normalement deux à cinq fois plus rapide qu'une mise en œuvre simple et directe de la définition. Il faut une grande attention et un codage habile et il est conseillé de faire le code de somme de contrôle "rapide comme l'éclair". Voir [RFC1071].

(c) Code pour le cas général
Le traitement du protocole TCP peut être compliqué, mais pour la plupart des segments il n'y a que quelques décisions simples à prendre. Le traitement par segment sera grandement accéléré par le codage de la ligne principale de façon à minimiser le nombre de décisions dans le cas le plus courant.

4.2.4 Interface de couche application/TCP

4.2.4.1 Rapports asynchrones

Il DOIT y avoir un mécanisme de rapport des conditions d'erreur douce de TCP à l'application. En général, on suppose que cela prend la forme d'un sous-programme ERROR_REPORT fourni par l'application qui peut être invoqué de façon asynchrone à partir de la couche transport [RFC0817] :

ERROR_REPORT(nom de la connexion locale, cause, sous-cause)

Le codage précis des paramètres cause et sous-cause n'est pas spécifié ici. Cependant, les conditions qui sont rapportées en asynchrone à l'application DOIVENT inclure :

* un message d'erreur ICMP est arrivé (voir le paragraphe 4.2.3.9)
* retransmissions excessives (voir le paragraphe 4.2.3.5)
* avance du pointeur d'urgence (voir le paragraphe 4.2.2.4).

Cependant, un programme d'application qui ne veut pas recevoir de tels appels ERROR_REPORT DEVRAIT être capable de désactiver effectivement ces appels.

Discussion :
Ces rapports d'erreur reflètent généralement des erreurs douces qui peuvent être ignorées sans dommage par de nombreuses applications. Il a été suggéré que ces appels de rapport d'erreur soient "désactivés" par défaut, mais ce n'est pas exigé.

4.2.4.2 Type-de-service

La couche application DOIT être capable de spécifier le Type-de-service (TOS) pour les segments qui sont envoyés sur une connexion. Bien que ce ne soit pas exigé, l'application DEVRAIT être capable de changer le TOS durant la durée de vie de

la connexion. TCP DEVRAIT passer la valeur courante du TOS sans changement à la couche IP lorsqu'il envoie des segments sur la connexion.

Le TOS sera spécifié indépendamment dans chaque direction de la connexion, de sorte que l'application receveuse spécifiera le TOS utilisé pour les segments ACK.

TCP PEUT passer le TOS le plus récemment reçu à l'application.

Discussion :
Certaines applications (par exemple, SMTP) changent la nature de leur communication durant la durée de vie d'une connexion, et voudraient donc changer la spécification du TOS.

Noter aussi que l'appel OPEN spécifié dans la RFC-793 inclut un paramètre ("options") dans lequel l'appelant peut spécifier des options IP telles que route de source, route enregistrée, ou horodatage.

4.2.4.3 Commande de purge

Certaines mises en œuvre de TCP ont inclus une commande FLUSH *(purge)*, qui va vider la file d'attente d'envoi de TCP de toutes les données pour lesquelles l'utilisateur a produit des commandes SEND mais qui sont toujours sur la droite de la fenêtre d'envoi en cours. C'est-à-dire qu'elle purge autant de données en file d'attente d'envoi que possible sans perdre la synchronisation des numéros de séquence. Ceci est utile pour la mise en œuvre de la fonction "interruption de sortie" de Telnet.

4.2.4.4 Rattachement multiple

L'interface d'utilisateur mentionnée aux paragraphes 2.7 et 3.8 de la RFC-793 a besoin d'être étendue pour le rattachement multiple. La commande OPEN DOIT avoir un paramètre facultatif :

OPEN(... [adresse IP locale,] ...)

pour permettre la spécification de l'adresse IP locale.

Discussion :
Certaines applications fondées sur TCP ont besoin de spécifier l'adresse IP locale à utiliser pour ouvrir une connexion particulière ; FTP en est un exemple.

Mise en œuvre :
Une commande OPEN passive avec un paramètre "adresse IP locale" spécifié va attendre une demande de connexion entrante à cette adresse. Si le paramètre est non spécifié, un OPEN passif attend une demande de connexion entrante pour n'importe quelle adresse IP locale, puis lie l'adresse IP locale de la connexion à l'adresse particulière qui est utilisée.

Pour une commande OPEN active, un paramètre "adresse IP locale" spécifiée sera utilisée pour ouvrir la connexion. Si le paramètre est non spécifié, le logiciel de réseautage va choisir une adresse IP locale appropriée (voir au paragraphe 3.3.4.2) pour la connexion

4.2.5 Résumé des exigences pour TCP

Caractéristique	**Parag.**	DOIT	DEVRAIT	PEUT	NE DEVRAIT PAS	NE DOIT PAS	Note
Fanion Push							
Agréger ou mettre en file d'attente des données non poussées	4.2.2.2			x			
L'émetteur compresse les fanions PSH successifs	4.2.2.2		x				
La commande SEND peut spécifier PUSH	4.2.2.2			x			
Sinon : l'émetteur met indéfiniment en mémoire tampon	4.2.2.2					x	

Sinon : PSH du dernier segment	4.2.2.2	x					
Notifier la réception ALP de PSH	4.2.2.2			x			1
Envoi de taille max de segment quand possible	4.2.2.2		x				
Fenêtre							
Traiter comme nombre non signé	4.2.2.3	x					
Traiter comme nombre de 32 bits	4.2.2.3		x				
Rétrécir la fenêtre à partir de la droite	4.2.2.16				x		
Robustesse contre la réduction de fenêtre	4.2.2.16	x					
Fenêtre du receveur close indéfiniment	4.2.2.17			x			
L'émetteur vérifie la fenêtre zéro	4.2.2.17	x					
Première vérif. après RTO	4.2.2.17		x				
Temporisation exponentielle	4.2.2.17		x				
Permet que la fenêtre reste indéfiniment à zéro	4.2.2.17	x					
L'émetteur accepte la fin de tempo. de conn. avec fenêtre zéro	4.2.2.17					x	
Données urgentes							
Pointeur pointe sur le dernier octet	4.2.2.4	x					
Séquence de données urgentes de longueur arbitraire	4.2.2.4	x					
Informer ALP en async. de données urgentes	4.2.2.4	x					1
ALP peut apprendre si/combien de donn. urg. en file d'attente	4.2.2.4	x					1
Options TCP							
Reçoit l'option TCP dans tout segment	4.2.2.5	x					
Ignore les options non prises en charge	4.2.2.5	x					
S'accomode de la longueur d'option illégale	4.2.2.5	x					
Accepte envoi/réception d'option MSS	4.2.2.6	x					
Envoie option MSS sauf si 536	4.2.2.6		x				
Envoie toujours option MSS	4.2.2.6			x			
MSS d'envoi par défaut est 536	4.2.2.6	x					
Calcule taille effective de segment d'envoi	4.2.2.6	x					
Somme de contrôle TCP							
L'émetteur calcule la somme de contrôle	4.2.2.7	x					
Le receveur vérifie la somme de contrôle	4.2.2.7	x					
Utilise la sélection ISN par horloge	4.2.2.9	x					
Ouverture de connexions							
Accepte les tentatives d'ouverture simultanées	4.2.2.10	x					
SYN-RCVD se souvient du dernier état	4.2.2.11	x					
Commande ouvert. passive interf. avec autres	4.2.2.18					x	
Les LISTEN fonct. simult. sur même accès	4.2.2.18	x					
Demande à IP adresse de src pour SYN si néc.	4.2.3.7	x					
Autrement, utilise adr. locale de connexion	4.2.3.7	x					
OPEN pour adr. IP diff/diffusion groupée	4.2.3.14					x	
Éliminer en silence seg pour adr. dif/dif. group	4.2.3.14	x					
Clôture de connexions							
RST peut contenir des données	4.2.2.12		x				
Informe applic. d'interruption de connexion	4.2.2.13	x					
Clôture connexions semi-duplex	4.2.2.13			x			
Envoie RST pour indiquer perte de données	4.2.2.13		x				
En état TIME-WAIT pour 2xMSL secondes	4.2.2.13	x					
Accepte SYN de l'état TIME-WAIT	4.2.2.13			x			
Retransmissions							
Algorithme Jacobson de démarrage lent	4.2.2.15	x					
Algorithme Jacobson d'évit. d'encombrement	4.2.2.15	x					
Retransmet avec même identifiant IP	4.2.2.15			x			
Algorithme de Karn	4.2.3.1	x					
Alg. Jacobson d'estimation de RTO	4.2.3.1	x					
Temporisation exponentielle	4.2.3.1	x					

Calcul de SYN RTO comme pour les données	4.2.3.1		x				
Valeurs et limites initiales recommandées	4.2.3.1		x				
Génération des ACK :							
Mise en file d'attente des segments décalés	4.2.2.20		x				
Traite toute la file d'attente avant envoi ACK	4.2.2.20	x					
Envoi de ACK pou segment décalé	4.2.2.21			x			
ACK retardés	4.2.3.2		x				
Retard < 0,5 seconde	4.2.3.2	x					
ACK tous les 2nd segments de taille complète	4.2.3.2	x					
Algorithme évitement SWS du receveur	4.2.3.3	x					
Envoi des données							
TTL configurable	4.2.2.19	x					
Algorithme évitement SWS de l'émetteur	4.2.3.4	x					
Algorithm e de Nagle	4.2.3.4		x				
L'application peut désactiver alg. Nagle	4.2.3.4	x					
Défaillance de connexion :							
Avis négatif à IP sur retrans R1	4.2.3.5	x					
Clôture connexion sur retrans R2	4.2.3.5	x					
ALP peut régler R2	4.2.3.5	x					1
Informe ALP de R1≤retrans<R2	4.2.3.5		x				1
Valeurs recommandées pour R1, R2	4.2.3.5		x				
Même mécanisme pour les SYN	4.2.3.5	x					
R2 d'au moins 3 minutes pour SYN	4.2.3.5	x					
Envoi des paquets Garder-en-vie :	4.2.3.6			x			
- L'application peut le demander	4.2.3.6	x					
- Désactivé par défaut	4.2.3.6	x					
- Envoi seulement si au repos pour intervalle	4.2.3.6	x					
- Intervalle configurable	4.2.3.6	x					
- Par défaut au moins 2 h.	4.2.3.6	x					
- Tolérant à la perte de ACK	4.2.3.6	x					
Options IP							
Ignore les options non comprises de TCP	4.2.3.8	x					
Accepte horodatage	4.2.3.8			x			
Accepte Route enregistrée	4.2.3.8			x			
Route de source :							
ALP peut la spécifier	4.2.3.8	x					1
Subroge la route de src dans le datagramme	4.2.3.8	x					
Construit route de retour d'après route de src	4.2.3.8	x					
Dernière route de source prend le pas	4.2.3.8		x				
Réception de messages ICMP de IP	4.2.3.9	x					
Dest. injoign (0,1,5) => informe ALP	4.2.3.9		x				1
Dest. injoign (0,1,5) => interrompt connexion	4.2.3.9					x	
Dest. injoign (2-4) => interropt connexion	4.2.3.9		x				
Source éteinte => démarrage lent	4.2.3.9		x				
Temps expiré => le dire à ALP, ne pas interr.	4.2.3.9		x				1
rob. param. => le dire à ALP, ne pas intrpre	4.2.3.9		x				1
Validation d'adresse							
Rejet commande OPEN pour adresse IP inval	4.2.3.10	x					
Rejet SYN venant d'adresse IP invalide	4.2.3.10	x					
Élimine en silence SYN pour adr dif/dif group	4.2.3.10	x					
Services d'interface TCP/ALP							1
Mécanisme de rapport d'erreur	4.2.4.1	x					
ALP peut désactiver routine rapport d'erreur	4.2.4.1		x				1
ALP peut spécifier TOS pour l'envoi	4.2.4.2	x					1
Passé inchangé à IP	4.2.4.2		x				
ALP peut changer TOS durant connexion	4.2.4.2		x				1
Passe TOS reçu à ALP	4.2.4.2			x			1
Commande FLUSH	4.2.4.3			x			

Paramètre d'adr. IP locale facult. dans OPEN	4.2.4.4	x					

Note : (1) "ALP" signifie programme de couche application *(Application Layer Program)*

5 Références

[DDN-NIC] "DDN Protocol Handbook," NIC-50004, NIC-50005, NIC-50006, (trois volumes), SRI International,

[INTRO:8] "The Structuring of Systems Using Upcalls," D. Clark, 10ème ACM SOSP, Orcas Island, Washington, décembre 1985.

[INTRO:9] "A Protocol for Packet Network Intercommunication," V. Cerf and R. Kahn, IEEE Transactions on Communication, mai 1974.

[INTRO:10] "The ARPA Internet Protocol," J. Postel, C. Sunshine, and D. Cohen, Computer Networks, Vol. 5, No. 4, juillet 1981.

[INTRO:11] "The DARPA Internet Protocol Suite," B. Leiner, J. Postel, R. Cole and D. Mills, Proceedings INFOCOM 85, IEEE, Washington DC, mars 1985. Et dans : IEEE Communications Magazine, mars 1985. Aussi disponible sous la référence ISI-RS-85-153.

[IP:9] "Fragmentation Considered Harmful," C. Kent and J. Mogul, ACM SIGCOMM-87, août 1987. Publié dans ACM Comp Comm Review, Vol. 17, n° 5. Cet utile papier expose les problèmes créés par la fragmentation Internet et présente des solutions de remplacement.

[MIL.1777] "Norme du protocole Internet militaire" MIL-STD-1777, Département de la Défense, août 1983. Cette spécification, telle qu'amendée par la RFC-963, est destinée à décrire le protocole Internet mais souffre de sérieuses omissions (par exemple, l'extension obligatoire de sous-réseau [RFC0950] et l'extension facultative de diffusion groupée [RFC1112]). Elle est aussi dépassée. En cas de conflit, les RFC-791, RFC-792, et RFC-950 doit être considérées comme ayant autorité, alors que le présent document est la référence en dernier ressort.

[RFC0768] J. Postel, "Protocole de datagramme d'utilisateur (UDP)", (STD 6), 28 août 1980.

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du protocole du programme Internet", STD 5, septembre 1981.

[RFC0792] J. Postel, "Protocole du message de contrôle Internet – Spécification du protocole du programme Internet DARPA", STD 5, septembre 1981. *(MàJ par la* RFC6633)

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

[RFC0813] D. Clark, "Fenêtre et stratégie d'accusé de réception dans TCP", juillet 1982. *(Historique)*

[RFC0814] D. Clark, "Noms, adresses, accès et chemins", juillet 1982. *(Information)*

[RFC0815] D. Clark, "Algorithmes de réassemblage de datagramme IP", juillet 1982.

[RFC0816] D. Clark, "Isolement et récupération de faute", juillet 1982. *(Historique)*

[RFC0817] D. Clark, "Modularité et efficacité dans une mise en œuvre de protocole", juillet 1982. *(Information)*

[RFC0826] D. Plummer, "Protocole de résolution d'adresses Ethernet : conversion des adresses de protocole réseau en adresses Ethernet à 48 bits pour transmission sur un matériel Ethernet", STD 37, novembre 1982.

[RFC0879] J. Postel. "Taille maximum de segment TCP et questions qui s'y rapportent", novembre 1983. *(Historique)*

[RFC0893] S. Lekkler et M. Karels, "Encapsulations d'en-queues", avril 1984.

[RFC0894] C. Hornig, "Norme pour la transmission des datagrammes IP sur les réseaux Ethernet", STD 41, avril 1984.

[RFC0896] J. Nagle, "Contrôle de l'encombrement dans l'inter-réseau IP/TCP", janvier 1984. *(Historique)*

[RFC0922] J. Mogul, "Diffusion des datagrammes Internet en présence de sous-réseaux", octobre 1984.

[RFC0950] J. Mogul et J. Postel, "Procédure standard de sous-réseautage Internet", (STD 5) août 1985.

[RFC0963] "Quelques problèmes de la spécification de la norme du protocole Internet militaire" D. Sidhu, RFC-963, novembre 1985.

[RFC0964] D. Sidhu, "Quelques problèmes de la spécification de la norme du protocole de contrôle de transmissions militaires", novembre 1985 *(Information)*

[RFC0980] O. Jacobsen et J. Postel, "Comment se procurer les documents de protocole (les RFC)", mars 1986.

[RFC0994] Organisation internationale de normalisation, "Texte final de la norme internationale 8473, Protocole pour la fourniture de service réseau en mode sans connexion", mars 1986.

[RFC0995] Organisation internationale de normalisation, "Protocole d'échange d'informations d'acheminement entre système d'extrémité et système intermédiaire à utiliser conjointement avec la norme ISO 8473", avril 1986.

[RFC1009] R. Braden et J. Postel, "Exigences pour les routeurs de l'Internet", juin 1987. *(Obsolète, voir* RFC1812*) (Historique)*

[RFC1011] J. Reynolds et J. Postel, "Protocoles officiels de l'Internet", mai 1987.

[RFC1016] W. Prue et J. Postel, "Ce qu'un hôte peut faire avec l'extinction de source : retard introduit par l'extinction de source (SQUID)", juillet 1987. Cette RFC décrivait surtout les adresses de diffusion dirigée. Cependant, le corps de la RFC concerne les routeurs, pas les hôtes.

[RFC1042] J. Postel et J. Reynolds, "Norme pour la transmission des datagrammes IP sur les réseaux IEEE 802", février 1988. (STD 43). Cette RFC contient un grand nombre d'informations d'importance pour les mises en œuvre Internet qui projettent d'utiliser les réseaux IEEE 802.

[RFC1071] R. Braden, D. Borman et C. Partridge, "Calcul de la somme de contrôle Internet", septembre 1988. *(Information) (Mise à jour par les RFC 1141 et 1624.)*

[RFC1072] V. Jacobson et R. Braden, "Extensions TCP pour les chemins à fort délai", octobre 1988. *(Obsolète, voir 1323 et 2018)*

[RFC1108] S. Kent, "Options de sécurité du Ministère US de la défense pour le protocole Internet", novembre 1991. *(Historique)*

[RFC1112] S. Deering, "Extensions d'hôte pour diffusion groupée sur IP", STD 5, août 1989. *(Mise à jour par la* RFC2236*)*

[RFC1123] R. Braden, éditeur, "Exigences pour les hôtes Internet – Application et prise en charge", STD 3, octobre 1989. *(MàJ par* RFC7766)

[RFC1700] J. Reynolds et J. Postel, "Numéros alloués", STD 2, octobre 1994. *(Historique, voir www.iana.org))*

[TCP:2] "Protocole de commande de transmission" MIL-STD-1778, US Department of Defense, août 1984. Cette spécification telle qu'amendée par la RFC-964 est destinée à décrire le même protocole que la [RFC0793]. En cas de conflit, la RFC-793 prend le pas, et le présent document a autorité sur les deux.

[TCP:6] P. Karn & C. Partridge, "Estimation du délai d'aller retour" ACM SIGCOMM-87, août 1987.

[TCP:7] "Évitement et contrôle d'encombrement," V. Jacobson, ACM SIGCOMM-88, août 1988.

Considérations pour la sécurité

Il y a de nombreuses questions concernant la sécurité dans les couches de communication du logiciel d'hôte, mais un exposé complet sort du domaine d'application de la présente RFC.

L'architecture Internet fournit généralement peu de protection contre l'usurpation des adresses de source IP, aussi tout mécanisme de sécurité qui se fonde sur la vérification de l'adresse IP de source d'un datagramme devrait être traité avec prudence. Cependant, dans des environnements restreints, certaines vérifications d'adresse de source peuvent être possibles. Par exemple, il peut y avoir un LAN sécurisé dont les passerelles avec le reste de l'Internet éliminent tout datagramme entrant avec une adresse de source qui usurpe l'adresse du LAN. Dans ce cas, un hôte sur le LAN pourrait utiliser l'adresse de source pour confronter la source locale et la source distante. Ce problème est compliqué par l'acheminement de source, et certains ont suggéré que la transmission par les hôtes de datagrammes à acheminement de source (voir au paragraphe 3.3.5) devrait être interdite pour des raisons de sécurité.

Les questions en rapport avec la sécurité sont mentionnés aux paragraphes concernant l'option de sécurité IP (paragraphe 3.2.1.8), le message ICMP Problème de paramètre (paragraphe 3.2.2.5), options IP dans les datagrammes UDP (paragraphe 4.1.3.2), et accès TCP réservés (paragraphe 4.2.2.1).

Adresse de l'auteur

Robert Braden
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
téléphone : (213) 822 1511
mél : Braden@ISI.EDU