Degouet Fabien
Desgeorge Guillaume
Corbel Steve

ESEO

Recherche bibliographique
Synthèse de protocoles courants

Année 98/99

I3 Réseaux et télécommunications

INTRODUCTION

Le but des réseaux est de faire communiquer plusieurs ordinateurs ensemble. Si les hommes communiquent entre eux grâce aux différentes langues, les ordinateurs utilisent différents protocoles. Les communications sont souvent internationales, et comme pour les hommes, il n'existe pas de protocole universel. Certains sont plus utilisés que d'autres, il en existe cependant un très grand nombre, chacun cherchant à imposer sa propre norme. Par soucis de clarté, nous n'étudierons dans ce rapport que les protocoles les plus courants.

Comment expliquer clairement ce qu'est un protocole? Supposons que quelqu'un veuille envoyer une lettre à quelqu'un d'autre. On va placer cette lettre dans une enveloppe et on y notera l'adresse. Pour l'acheminement du courrier, le contenu de la lettre n'est d'aucune utilité. Les différents services de la poste regardent les différents champs de l'adresse et dirigent l'enveloppe, donc son contenu dans la bonne direction.

Il en est de même quand un ordinateur veut envoyer des données à un autre ordinateur. Les données sont enfermées (on dit encapsulées) dans une enveloppe qui contient les informations permettant l'acheminement des données. Un protocole, c'est la façon dont l'adresse est écrite sur l'enveloppe, le fait de mettre d'abord le nom, puis la rue et enfin la ville. Un autre protocole, c'est aussi le fait de mettre le lieu et la date en haut à droite et la signature en bas.

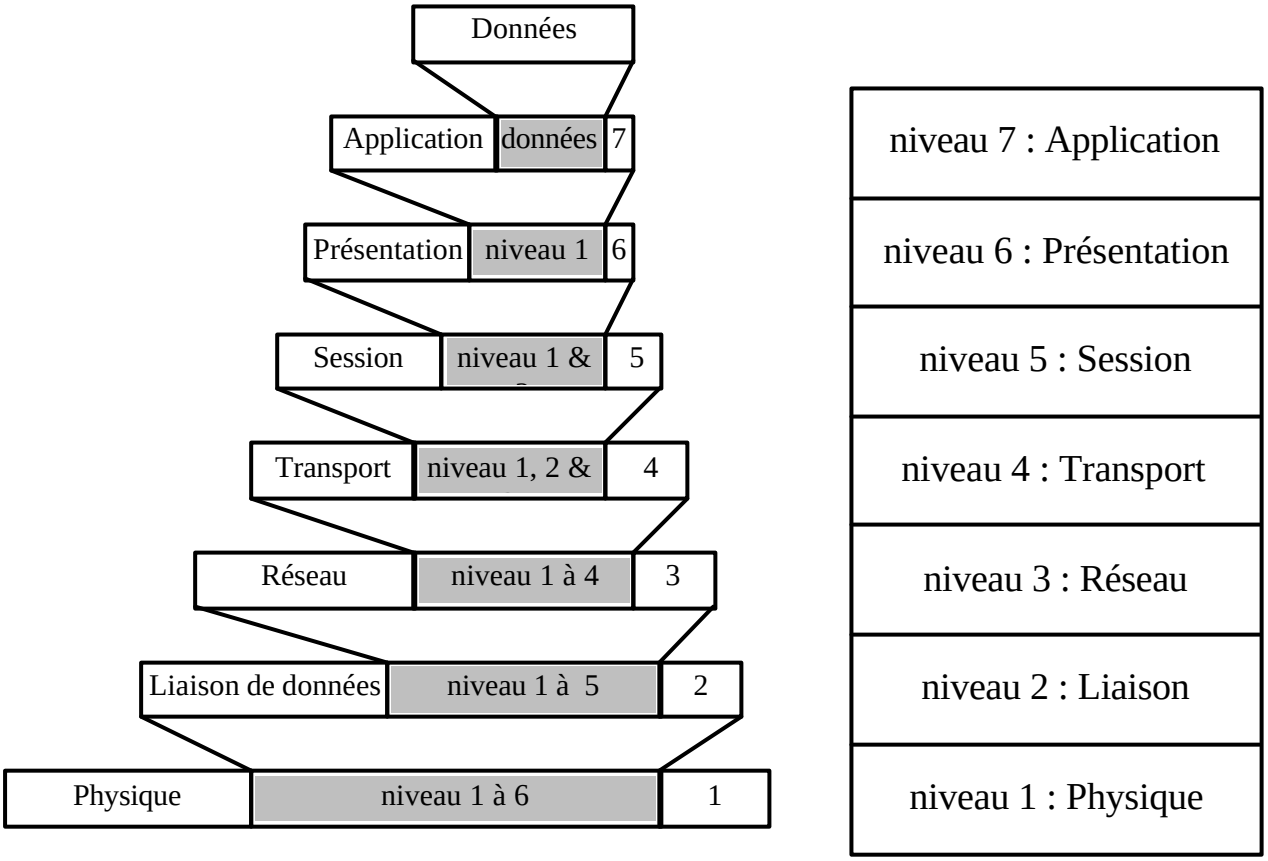
Finalement, un protocole est une description formelle de règles et de conventions à suivre dans un échange d'informations, que ce soit pour acheminer les données jusqu'au destinataire ou pour que le destinataire comprenne comment il doit utiliser les données qu'il a reçues.

Pour s'y retrouver plus facilement dans l'ensemble des protocoles, l'International Standard Organization (ISO) a défini un modèle de base appelé modèle OSI. Ce modèle définit 7 niveaux différents pour le transfert de données. Ces niveaux sont également appelés couches.

Le septième niveau, la couche Application, gère le transfert des informations entre programmes. Le sixième niveau, la couche Présentation, s'occupe de la mise en forme des textes et des conventions d'affichage. Le cinquième niveau, la couche Session, s'occupe de l'établissement, de la gestion et coordination des communications. Le quatrième niveau, la couche Transport, gère la remise correcte des informations. Vient ensuite le niveau trois, la couche Réseau, qui détermine les routes de transport et qui s'occupe du traitement et du transfert de messages. Le niveau deux, la couche Liaison de données, s'occupe du codage, de l'adressage, et de la transmission des informations. Le premier niveau, la couche physique, gère les connections matérielles.

A chacun de ces niveaux, on encapsule un en-tête et une fin de trame qui comporte les informations nécessaires en suivant les règles définies par le protocole utilisé. Sur le schéma ci-dessous, la partie qui est rajoutée à chaque niveau est la partie sur fond blanc. La partie sur fond grisé est celle obtenue après encapsulation du niveau précédent. La dernière trame, celle qu'on obtient après avoir encapsulé la couche physique, est celle qui sera envoyée sur le réseau.

Le modèle OSI



LA FAMILLE TCP/IP

I. TCP : TRANSMISSION CONTROL PROTOCOL

1. INTRODUCTION

Le protocole TCP est défini dans le but de fournir un service de transfert de données de haute fiabilité entre deux ordinateurs "maîtres" raccordés sur un réseau de type "paquets commutés", et sur tout système résultant de l'interconnexion de ce type de réseaux.

1.Motivation

La communication entre systèmes d'information joue un rôle croissant dans les domaines militaires, institutionnels, scientifiques et commerciaux.
Au fur et à mesure que les réseaux de communication informatiques à caractère stratégiques ou tactiques sont déployés, il devient essentiel de trouver un moyen d'interconnexion de ces réseaux, et des standards de transmission de données permettant de supporter une vaste gamme d'applications. Anticipant le besoin de tels standards, le député et sous-secrétaire d'état à la recherche de la Défense Américaine a officialisé le protocole décrit ici en tant que base pour la standardisation des processus d'intercommunication de données du Département de la Défense Américaine (DoD).

TCP est un protocole sécurisé orienté connexion conçu pour s'implanter dans un ensemble de protocoles multicouches, supportant le fonctionnement de réseaux hétérogènes. TCP fournit un moyen d'établir une communication fiable entre deux tâches exécutées sur deux ordinateurs autonomes raccordés à un réseau de données. Le protocole TCP s'affranchit le plus possible de la fiabilité intrinsèques des couches inférieures de communication sur lesquelles il s'appuie. TCP suppose donc uniquement que les couches de communication qui lui sont inférieures lui procurent un service de transmission de paquet simple, dont la qualité n'est pas garantie.

En principe, TCP doit pouvoir supporter la transmission de données sur une large gamme d'implémentations de réseaux, depuis les liaisons filaires câblées, jusqu'aux réseaux commutés, ou asynchrones.

TCP s'intègre dans une architecture multicouche des protocoles, juste au-dessus du protocole Internet IP. Ce dernier permet à TCP l'envoi et la réception de segments de longueur variable, encapsulés dans un paquet Internet appelé aussi "datagramme". Le datagramme Internet dispose des mécanismes permettant l'adressage d'un service TCP source et un destinataire, quelles que soient leur position dans le réseau. Le protocole IP s'occupe aussi de la fragmentation et du réassemblage des paquets TCP lors de la traversée de réseaux de plus faibles caractéristiques. Le protocole IP transporte aussi les informations de priorité, compartimentation et classification en termes de sécurité relatives aux segments TCP. Ces informations se retrouvent alors transmises de bout en bout de la communication.

Couches de protocoles

Niveaux Supérieurs
TCP
IP
Couche Physique

De grandes parties de ce document sont écrites dans un contexte où les implémentations TCP sont concomitantes à d'autres protocoles de haut niveau dans la même machine. Certains systèmes informatiques seront raccordés au réseau via un frontal qui accueillera les fonctions TCP et IP, ainsi que les protocoles réseau de bas niveau. La spécification TCP décrit une interface à destination des applications de niveau supérieur, y compris dans le cas d'une architecture avec un frontal, pour autant que les protocoles "poste vers frontal" soient implémentés.

1.2. Portée

TCP prétend fournir un service de communication de processus à processus, dans un environnement réseau complexe. TCP est défini comme un protocole de communication "host to host", c'est à dire de maître à maître (par opposition à "central à terminal").

2. SPECIFICATION FONCTIONNELLE

2.1. Format de l'en-tête

Les paquets TCP sont envoyés sous forme de datagrammes Internet. L'en-tête IP transmet un certain nombre de paramètres, tels que les adresses Internet source et destinataires.

L'en-tête TCP est placée à la suite, contenant les informations spécifiques au protocole TCP. Cette division permet l'utilisation de protocoles autres que TCP, au dessus de la couche IP.

En-tête TCP

0 16 32bits

<table>
<tr><td colspan="8">Port Source</td><td colspan="8">Port Destination</td></tr>
<tr><td colspan="16">Numéro de séquence</td></tr>
<tr><td colspan="16">Accusé de réception</td></tr>
<tr><td colspan="4">Data Offset</td><td colspan="4">Réservé</td><td>U</td><td>A</td><td>P</td><td>R</td><td>S</td><td>F</td><td colspan="4">Fenêtre</td></tr>
<tr><td colspan="8">Checksum</td><td colspan="8">Pointeur données urgentes</td></tr>
<tr><td colspan="8">Option</td><td colspan="8">Bourrage</td></tr>
<tr><td colspan="16">Data</td></tr>
</table>

Port source: 16 bits

Le numéro de port de la source.

Port Destinataire: 16 bits

Le numéro de port du destinataire.

Numéro de séquence: 32 bits

Le numéro du premier octet de données par rapport au début de la transmission (sauf si SYN est marqué). Si SYN est marqué, le numéro de séquence est le numéro de séquence initial (ISN) et le premier octet à pour numéro ISN+1.

Accusé de réception: 32 bits

Si ACK est marqué ce champ contient le numéro de séquence du prochain octet que le récepteur s'attend à recevoir. Une fois la connexion établie, ce champ est toujours renseigné.

Data Offset: 4 bits

La taille de l'en-tête TCP en nombre de mots de 32 bits. Il indique là ou commence les données. L'en-tête TCP, dans tous les cas à une taille correspondant à un nombre entier de mots de 32 bits.

Réservé: 6 bits

Réservés pour usage futur. Doivent nécessairement être à 0.

Bits de contrôle: 6 bits (de gauche à droite):

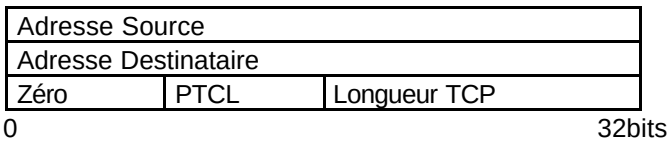
URG: Pointeur de données urgentes significatif
ACK: Accusé de réception significatif
PSH: Fonction Push
RST: Réinitialisation de la connexion
SYN: Synchronisation des numéros de séquence
FIN: Fin de transmission

Fenêtre: 16 bits

Le nombre d'octets à partir de la position marquée dans l'accusé de réception que le récepteur est capable de recevoir.

Checksum: 16 bits

Le Checksum est constitué en calculant le complément à 1 sur 16 bits de la somme des compléments à 1 des octets de l'en-tête et des données pris deux par deux (mots de 16 bits). Si le message entier contient un nombre impair d'octets, un 0 est ajouté à la fin du message pour terminer le calcul du Checksum. Cet octet supplémentaire n'est pas transmis. Lors du calcul du Checksum, les positions des bits attribués à celui-ci sont marqués à 0.
Le Checksum couvre de plus une pseudo en-tête de 96 bits préfixée à l'en-tête TCP. Cette pseudo en-tête comporte les adresses Internet source et destinataires, le type de protocole et la longueur du message TCP. Ceci protège TCP contre les erreurs de routage. Cette information sera véhiculée par IP, et est donnée comme argument par l'interface TCP/Réseau lors des appels d'IP par TCP.



La longueur TCP compte le nombre d'octets de l'en-tête TCP et des données du message, en excluant les 12 octets de la pseudo en-tête.

Pointeur de données urgentes: 16 bits

Communique la position d'une donnée urgente en donnant son décalage par rapport au numéro de séquence. Le pointeur doit pointer sur l'octet suivant la donnée urgente. Ce champs n'est interprété que lorsque URG est marqué.

Options: variable

Les champs d'option peuvent occuper un espace de taille variable à la fin de l'en-tête TCP. Ils formeront toujours un multiple de 8 bits. Toutes les options sont prises en compte par le Checksum. Un paramètre d'option commence toujours sur un nouvel octet. Il est défini deux formats types pour les options:

Cas 1: Option mono-octet.

Cas 2: Octet de type d'option, octet de longueur d'option, octets de valeurs d'option.

La longueur d'option prend en compte l'octet de type, l'octet de longueur lui-même et tous les octets de valeur et est exprimée en octets.

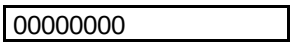
Notez que la liste d'option peut être plus courte que ce que l'offset de données pourrait le faire supposer. Un octet de remplissage (padding) devra être dans ce cas rajouté après le code de fin d'options. Ce octet est nécessairement à 0.

TCP doit implémenter toutes les options.

Actuellement, les options définies sont (type indiqué en octal):

Type	Longueur	Description
0	-	Fin de liste d'option
1	-	Nop
2	4	Taille de segment maximal

Définition des options spécifiques :
Fin de liste d'options



Type=0

Ce code indique la fin du champ d'options. Sa position peut ne pas coïncider avec l'indication du début du champ de données marqué dans l'Offset de données. Il doit être placé après toutes les options, et non après chaque option. Il ne doit être utilisé que dans le cas ou la fin des options ne coïncide pas avec le début du champ de données.

No-Operation

Type=1

Cette option peut être utilisée entre deux options, par exemple pour aligner le début d'une option sur un début de mot de 16 bits. L'utilisation de ce séparateur n'est pas une obligation. L'implémentation doit donc prévoir de pouvoir prendre en compte un option même au milieu d'un mot.

Taille maximale de segment

Type=2 Longueur=4

Donnée d'option : Taille maximale de segment: 16 bits

Si cette option est présente, elle communique à l'émetteur la taille maximale des segments qu'il pourra envoyer. Ce champ doit être envoyé dans la requête de connexion initiale (avec SYN marqué). Si cette option est absente, le segment pourra être pris de n'importe quelle taille.

Bourrage (padding): variable

Les octets de bourrage terminent l'en-tête TCP:

de sorte que le nombre d'octet de celle-ci soit toujours multiple de 4 (32 bits)
de sorte que l'offset de données marqué dans l'en-tête corresponde bien au début des données applicatives.

3. Etablissement et rupture des connexions

TCP indique un identificateur de port. Comme ces identificateurs sont choisis indépendamment par chaque extrémité, ils peuvent se révéler identiques. L'adresse unique d'une communication TCP est obtenue par la concaténation de l'adresse Internet avec l'identificateur du port sélectionné, constituant ainsi ce que l'on nome une "socket". Cette socket est alors unique dans l'ensemble du réseau.

Une connexion de base est définie par un couple de sockets, l'un définissant l'émetteur, l'autre le récepteur. Un socket peut devenir le destinataire ou la source pour plusieurs sockets distinctes. La connexion est résolument bidirectionnelle, et prend la dénomination de "full-duplex".

TCP est libre d'associer ses ports avec les processus exécutés sur sa machine. Cependant, quelques règles ont été établies pour l'implémentation. Ont été définis un certain nombre de sockets "réservés" que TCP ne doit associer qu'avec certains processus bien identifiés. Ceci revient à dire que certains processus peuvent s'attribuer la propriété de certains ports, et ne pourront initier de communication que sur ceux-ci. (Actuellement, cette "propriété" est issue d'une implémentation locale, mais nous envisageons une commande utilisateur Request Port, ou une autre méthode pour assigner automatiquement un ensemble de ports à une application, par exemple en utilisant quelques bits de poids fort du numéro de port pour coder l'application).

Une connexion est demandée par activation de la commande OPEN indiquant le port local et les paramètres du socket distant. En retour, TCP répond par un nom local (court) symbolique que l'application utilisera dans ses prochains appels. Plusieurs choses doivent être retenues à propos des connexions. Pour garder la trace de cette connexion, nous supposons l'existence d'une structure de données appelée Transmission Control Block (TCB). Une des stratégies d'implémentation est de dire que le nom local donné est un pointeur vers le TCB associé à cette connexion. La commande OPEN spécifie en outre si le processus de connexion doit être effectué jusqu'à son terme, ou s'il s'agit d'une ouverture en mode passif.

Une ouverture passive signifie que le processus de connexion se met en attente d'une demande de connexion plutôt que de l'initier lui-même. Dans la plupart des cas, ce mode est utilisé lorsque l'application est prête à répondre à tout appel. Dans ce cas, le socket distant spécifié n'est composé que de zéros (socket indéfini). Le socket indéfini ne peut être passé à TCP que dans le cas d'une connexion passive.

Un utilitaire désireux de fournir un service à un processus non identifié pourra initier une connexion passive. Tout appelant effectuant une requête de connexion sur le socket local sera reconnu. Il sera bon de garder en mémoire que ce socket est associé à ce service.
Les sockets "réservés" sont un bon moyen d'associer à priori des ports à des applications standard. Par exemple, le serveur "Telnet" est en permanence associé à un socket particulier, d'autres étant réservés pour les transferts de fichiers, sessions de terminal distant, générateur de texte, écho (ces deux pour des besoins de test), etc. Un socket peut être réservé à la fonction de serveur de domaines, transcodant les "noms explicites" de services en sockets Internet. Si le concept même de l'assignation à priori de sockets fait partie de TCP, l'assignation concrète des sockets "réservés" est définie dans un autre document.

Les processus peuvent ouvrir une connexion passive et attendre qu'une connexion active les impliquant provienne d'une autre machine. TCP aura la charge d'avertir l'application qu'une communication est établie. Deux processus émettant au même moment une requête de connexion l'un vers l'autre se retrouveront normalement connectés. Cette souplesse est indispensable pour assurer un bon fonctionnement du réseau composé d'éléments totalement asynchrones.

Les deux cas de conclusion d'une communication impliquant une connexion passive et une active sont les suivants. Soit le socket distant a été précisé lors de la requête de connexion passive, auquel cas seule une requête de connexion du distant attendu vers le local peut aboutir à l'établissement d'une

communication. Soit le socket distant a été laissé indéfini, et toute requête de connexion sur le socket local, d'où qu'elle vienne aboutit à une communication valide. D'autres fonctionnalités permettront une acceptation sur correspondance partielle entre sockets.

Si plusieurs requêtes de connexion passive sont en attente (enregistrées dans la table de TCBs) pour le même socket local, et qu'une demande de connexion active provient de l'extérieur, le protocole prévoit de d'abord chercher s'il l'une des requêtes dont le socket distant a été clairement exprimé correspond à celui de la demande. Si tel est le cas, ce socket sera activé. Sinon, c'est une requête "indéfinie" qui sera activée.

La procédure de connexion utilise le bit de contrôle de synchronisation (SYN) et suppose la transmission de trois messages. Cet échange est appelé "négociation ternaire".

La connexion suppose le rendez-vous d'un segment marqué du bit SYN et d'une requête locale (TCB), chacun des deux étant créé par l'exécution d'une commande de connexion. La correspondance entre le socket arrivé et le socket attendu détermine l'opportunité de la connexion. Celle-ci ne devient réellement établie que lorsque les deux numéros de séquence ont été synchronisés dans les deux directions.

La rupture d'une connexion suppose l'émission de segments, marqués du bit FIN.

4. Communication de données

Les données circulant dans la connexion ouverte doivent être vues comme un flux d'octets. L'application indique dans la commande SEND si les données soumises lors de cet appel (et toutes celles en attente) doivent être immédiatement émises par l'activation du flag PUSH.

Par défaut, TCP reste libre de stocker les données soumises par l'application pour les émettre à sa convenance, jusqu'à ce que le signal PUSH soit activé. Dans ce dernier cas, toutes les données non émises doivent être envoyées. Symétriquement, lorsque le TCP récepteur voit le flag PUSH marqué, il devra passer immédiatement toutes les données collectées à l'application destinataire.

Il n'y a à priori aucune corrélation entre la fonction PUSH et les limites des segments. Les données d'un segment peuvent être le résultat d'une seule commande SEND, en tout ou partie, ou celui de plusieurs appels SEND.

La fonction de la fonction push et du flag PUSH est de forcer la transmission immédiate de toutes les données latentes entre les deux TCP. Il ne s'agit aucunement d'une fonction d'enregistrement (Cf. langage Perl).

Il y a par contre une relation entre la fonction push et l'usage des tampons dans l'interface TCP/application. Chaque fois qu'un flag PUSH est associé à des données stockées dans le tampon de réception, celui-ci est intégralement transmis à l'application même s'il n'est pas plein. Si le tampon est rempli avant qu'un flag PUSH soit vu, les données sont transmises à l'application par éléments de la taille du tampon.

TCP dispose d'un moyen d'avertir l'application que, dans le flux de données qu'il est en train de lire, au delà de la position de lecture courante, des données de caractère urgent sont apparues. TCP ne définit pas ce que l'application est sensée faire lorsqu'elle est avisée de la présence de ces données. En général, c'est l'implémentation de l'application qui traitera ces données urgentes selon ses besoins propres.

5. Priorité et Sécurité

TCP utilise le champ "type de service" et les options de sécurité du protocole Internet pour fournir les fonctions relatives à la priorité et la sécurité des communications TCP, sur un principe de "détection". Tous les modules TCP ne fonctionneront pas nécessairement dans un environnement sécurisé à plusieurs niveaux; certains pourront être limités à un fonctionnement sans sécurité, d'autres ne pourront

prendre en compte qu'un seul niveau à la fois. Par conséquent, les implémentations TCP ne pourront répondre en termes de sécurité qu'à un sous ensembles de cas du modèle sécurisé multi-niveaux.

Les modules TCP opérant dans un environnement sécurisé à plusieurs niveaux devront correctement renseigner les segments sortants en termes de sécurité, niveau de sécurité, et priorité. De tels modules TCP doivent fournir aux applications supérieures telles que Telnet ou THP une interface leur permettant de spécifier ces paramètres.

II. LE PROTOCOLE IP

1. Description fonctionnelle

La fonction ou rôle du Protocole Internet est d'acheminer les datagrammes à travers un ensemble de réseaux interconnectés. Ceci est réalisé en transférant les datagrammes d'un module Internet à l'autre jusqu'à atteindre la destination. Les modules Internet sont des programmes exécutés dans des hôtes et des routeurs du réseau Internet. Les datagrammes sont transférés d'un module Internet à l'autre sur un segment particulier de réseau selon l'interprétation d'une adresse Internet. De ce fait, un des plus importants mécanismes du protocole Internet est la gestion de cette adresse Internet.

Lors de l'acheminement d'un datagramme d'un module Internet vers un autre, les datagrammes peuvent avoir éventuellement à traverser une section de réseau qui admet une taille maximale de paquet inférieure à celle du datagramme. Pour surmonter ce problème, un mécanisme de fragmentation est géré par le protocole Internet.

Adressage

Une distinction doit être faite entre noms, adresses, et chemins. Un nom indique ce que nous cherchons. Une adresse indique où cela se trouve. Un chemin indique comment y aboutir. Le protocole Internet s'occupe essentiellement des adresses. C'est à des protocoles de niveau plus élevé (ex., hôte-vers-hôte ou application) que revient la tâche de lier des noms à des adresses. Le module Internet déduit de l'adresse Internet une adresse réseau local. La tâche qui consiste à transcrire l'adresse de réseau local en termes de chemin (ex., sur un réseau local ou dans un routeur) revient au protocole de bas niveau.

Les adresses ont une longueur fixe de 4 octets (32 bits). Une adresse commence toujours par un numéro de réseau, suivi d'une adresse locale (appelée le champ "reste") codant l'adresse de l'hôte sur ce réseau. Il existe trois formats ou classes d'adresses Internet : pour la classe A, le bit de poids fort vaut zéro, les 7 bits suivants désignent le réseau, les derniers 24 bits désignent l'adresse locale de la machine; pour la classe B, les deux bits de poids fort valent 1 et 0, les 14 bits suivants désignent le réseau et les 16 derniers bits l'adresse locale de machine ; pour la classe C, les trois bits de poids fort forment le schème 110, les 21 bits suivants forment l'adresse réseau et les 8 derniers bits l'adresse locale.

La transcription d'adresse Internet en adresses de réseau local doit être sujette à quelques précautions ; un hôte physique unique peut abriter plusieurs adresses Internet distinctes comme s'il s'agissait de plusieurs hôtes indépendants. Certains hôtes peuvent disposer de plusieurs interfaces physiques (multi-homing).

De ce fait, il faudra pouvoir considérer le cas d'un hôte à plusieurs interfaces physiques chacune abritant plusieurs adresses Internet distinctes.

Des exemples de répartition d'adresses peuvent être trouvés dans "Address Mappings" (rfc 1060).

Fragmentation

La fragmentation du datagramme Internet devient nécessaire dès lors qu'un datagramme de grande taille arrive sur une portion de réseau qui n'accepte la transmission que de paquets plus courts.

Un datagramme Internet peut être spécifié "non fractionnable" Un tel datagramme Internet ne doit jamais être fragmenté quelques soient les circonstances. Si un datagramme Internet non fractionnable ne peut être acheminé jusqu'à sa destination sans être fragmenté, alors il devra être rejeté.

La fragmentation, la transmission et le réassemblage à travers un réseau local hors de vue d'un module de protocole Internet est appelée fragmentation Intranet .

Les procédures de fragmentation et réassemblage Internet doivent pouvoir "casser" un datagramme Internet en un nombre de "fragments" arbitraire et quelconque pourvu que le réassemblage soit possible. Le récepteur des fragments utilise le champ d'identification pour s'assurer que des fragments de plusieurs datagrammes ne puissent être mélangés. Le champ "Fragment Offset" indique au récepteur la position du fragment reçu dans le datagramme original. Les champs "Fragment Offset" et "Longueur Totale" déterminent la portion du datagramme original que représente le fragment. L'indicateur bit "Dernier Fragment" indique (lors de sa remise à zéro) au récepteur qu'il s'agit du dernier fragment. Ces champs véhiculent suffisamment d'information pour réassembler les datagrammes.

Le champ d'identification sert à distinguer les fragments d'un datagramme de ceux d'un autre datagramme. Le module Internet émetteur d'un datagramme Internet initialise le champ d'identification à une valeur qui doit être unique pour cette paire source-destination et pour ce protocole pendant toute la durée de transmission de ce datagramme. Le module Internet terminant l'émission d'un datagramme met le bit "Dernier Fragment" et le champ "Fragment Offset" à zéro.

Pour fragmenter un long datagramme, un module Internet (par exemple, dans un routeur), crée deux nouveaux datagrammes et copie le contenu des champs d'en-tête Internet originaux dans les deux nouvel en-têtes. Les données du datagramme original sont divisées en deux portions, la première d'une taille multiple de 8 octets (64 bit) (la taille de la seconde portion n'est donc pas nécessairement un multiple de 8 octets). Nous appellerons le nombre de blocs de 8 octets dans la première portion NBF (ou Nombre de Blocs du Fragment). La première portion de données est placée dans le premier des deux nouveaux datagramme, et le champ "Longueur Totale" est renseigné avec la taille de ce datagramme.
Le bit "Dernier Fragment" est basculé à 1. La seconde portion de données est placée dans le second des deux nouveaux datagrammes, et le champ "longueur totale" est renseigné avec la taille du second datagramme. Le bit "Dernier Fragment" est placé à la même valeur que celui du datagramme original. Le champ "Fragment Offset" du second datagramme constitué est renseigné avec la valeur du même champ du datagramme original plus NFB.

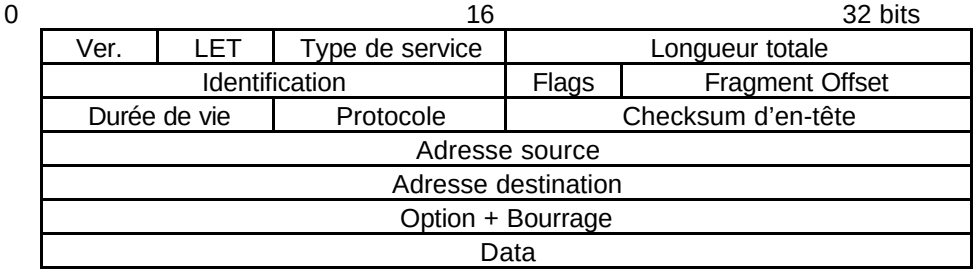
Cette procédure peut être généralisée à une fragmentation en n fragments, plutôt que les deux décrits ci-dessus.

Pour réassembler les fragments d'un datagramme Internet, un module Internet (par exemple dans un hôte destinataire) recombine les datagrammes dont les valeurs des quatre champs suivants sont identiques : identification, source, destination, et protocole. La recombinaison est réalisée en replaçant la portion de donnée contenue dans chaque fragment dans un tampon à la position relative indiquée par le champ "Fragment Offset" lu dans l'en-tête correspondant. Le premier fragment sera donc placé en début de tampon, et le dernier fragment récupéré aura le bit "Dernier Fragment" à zéro.

2. SPECIFICATION

2.1. Format d'en-tête Internet

Un résumé du contenu de l'en-tête Internet suit :



Version : 4 bits

Le champ Version renseigne sur le format de l'en-tête Internet. Ce document décrit le format de la version 4 du protocole.

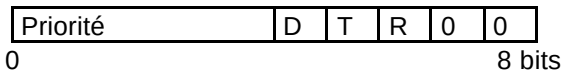
Longueur d'En-Tête : 4 bits

Le champ Longueur d'En-Tête (LET) code la longueur de l'en-tête Internet, l'unité étant le mots de 32 bits, et de ce fait, marque le début des données. Notez que ce champ ne peut prendre une valeur en dessous de 5 pour être valide.

Type de Service : 8 bits

Le Type de Service donne une indication sur la qualité de service souhaitée, qui reste cependant un paramètre "abstrait". Ce paramètre est utilisé pour "guider" le choix des paramètres des services actuels lorsqu'un datagramme transite dans un réseau particulier. Certains réseaux offrent un mécanisme de priorité, traitant préférentiellement un tel trafic par rapport à un trafic moins prioritaire (en général en acceptant seulement de véhiculer des paquets d'un niveau de priorité au dessus d'un certain seuil lors d'une surcharge momentanée). Principalement, le choix offert est une négociation entre les trois contraintes suivantes : faible retard, faible taux d'erreur, et haut débit.
Bits 0-2 :
Priorité.

Bit 3 :
0 = Retard standard,
1 = Retard faible.
Bits 4 :
0 = Débit standard,
1 = Haut débit.
Bits 5 :
0 = Taux d'erreur standard
1 = Taux d'erreur faible.
Bit 6-7 :
Réservé.



Priorité

111 -
Network Control
110 -
Internetwork Control
101 -
CRITIC/ECP
100 -
Flash Override
011 -
Flash
010 -
Immediate
001 -
Priority
000 -
Routine

L'utilisation des indications en termes de retard, débit, et qualité de transmission peut augmenter le "coût" (d'un certain point de vue) du service. Dans la plupart des réseaux, de meilleures performances pour l'un de ces paramètres s'obtient au prix d'une dégradation des performances pour un autre. A moins

d'une situation exceptionnelle, il sera préférable de ne pas activer plus de deux optimisations sur les trois.

Le "Type de Service" sert à préciser le traitement effectué sur le datagramme pendant sa transmission à travers Internet. Des exemples d'association de ce code aux améliorations de service proposées par des réseaux existants comme AUTODIN II, ARPANET, SATNET, et PRNET sont données dans la RFC 795 "Service Mappings" [8].

La priorité dite "Network Control" est stipulée comme étant une priorité à l'intérieur d'un seul réseau. Le fait d'utiliser cette option instaure une priorité pour chaque section traversée. La priorité "Internetwork Control" n'est gérée que par les routeurs. Si l'utilisation de ces priorités ont une signification particulière ou supplémentaire pour l'un des réseaux, il est de la responsabilité de ce dernier de lire et d'interpréter les présentes informations.

Longueur Totale : 16 bits

Le champ "Longueur Totale" est la longueur du datagramme entier y compris en-tête et données, mesurée en octets. Ce champ ne permet de coder qu'une longueur de datagramme d'au plus 65,535 octets. Une telle longueur rendrait de toutes façon les datagrammes impossible à gérer pour la plus grande partie des réseaux. Les hôtes devront au moins pouvoir accepter des datagrammes d'une longueur jusqu'à 576 octets (qu'il s'agisse d'un datagramme unique ou d'un fragment). Il est de même recommandé que des hôtes ne décident d'envoyer des datagrammes de plus de 576 octets que dans la mesure où ils sont sûrs que la destination est capable de les accepter.

Le nombre 576 a été choisi pour permettre à un bloc de données de taille raisonnable d'être transmis dans un datagramme, tenant compte des données à ajouter pour constituer les en-têtes de protocole. Par exemple, cette taille permet la transmission d'un bloc de 512 octets, plus 64 octets d'en-tête dans un datagramme unique. (NdT : je rappelle ici que la taille de 512 octets correspond à un secteur sur la plupart des supports de stockage) La taille maximale d'un en-tête Internet étant de 60 octets, et sa taille typique étant de 20 octets, ce nombre permet de conserver une bonne marge pour les données protocolaires de plus haut niveau.

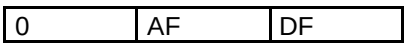
Identification : 16 bits

Une valeur d'identification assignée par l'émetteur pour identifier les fragments d'un même datagramme.

Flags : 3 bits

Divers commutateurs de contrôle.

Bit 0 : réservé, doit être laissé à zéro
Bit 1: (AF) 0 = Fragmentation possible,
1 = Non fractionnable.
Bit 2: (DF) 0 = Dernier fragment,
1 = Fragment intermédiaire.



Fragment Offset : 13 bits

Ce champ indique le décalage du premier octet du fragment par rapport au datagramme complet. Cette position relative est mesurée en blocs de 8 octets (64 bits). Le décalage du premier fragment vaut zéro.

Durée de vie : 8 bits

Ce champ permet de limiter le temps pendant lequel un datagramme reste dans le réseau. Si ce champ prend la valeur zéro, le datagramme doit être détruit. Ce champ est modifié pendant le traitement de l'en-tête Internet. La durée de vie est mesurée en secondes. Chaque module Internet doit retirer au moins une unité de temps à ce champ, même si le traitement complet du datagramme par le module est

effectué en moins d'une seconde. De ce fait, cette durée de vie doit être interprétée comme la limite absolue maximale de temps pendant lequel un datagramme peut exister. Ce mécanisme est motivé par la nécessité de détruire les datagrammes qui n'ont pu être acheminés, en limitant la durée de vie même du datagramme.

Protocole : 8 bits

Ce champ indique quel protocole de niveau supérieur est utilisé dans la section données du datagramme Internet. Les différentes valeurs admises pour divers protocoles sont listée dans la RFC "Assigned Numbers" [rfc1060].

Checksum d'en-tête : 16 bits

Un Checksum calculé sur l'en-tête uniquement. Comme certains champs de l'en-tête sont modifiés (ex., durée de vie) pendant leur transit à travers le réseau, ce Checksum doit être recalculé et vérifié en chaque point du réseau où l'en-tête est réinterprétée.

L'algorithme utilisé pour le Checksum est le suivant :

On calcule le complément à un sur 16 bits de la somme des compléments à un de tous les octets de l'en-tête pris par paires (mots de 16 bits). Lorsque l'on calcule le Checksum, on considère une en-tête dont le champ réservé pour ce même Checksum vaut zéro.

L'algorithme de Checksum peut paraître élémentaire mais l'expérimentation a montré que cette technique était suffisante. Il se peut que cet algorithme soit plus tard remplacé par un calcul de type CRC, suivant la nécessité future.

Adresse source : 32 bits

L'adresse Internet de la source.

Adresse destination : 32 bits

L'adresse Internet du destinataire.

Options : variable

Les datagrammes peuvent contenir des options. Celles-ci doivent être implémentées par tous les modules IP (hôtes et routeurs). Le caractère "optionnel" concerne leur transmission, et non leur implémentation.

Dans certains environnements, l'option de sécurité peut être obligatoire dans tous les datagrammes.

Le champ d'option est de longueur variable. Un datagramme peut comporter zéro ou plus options. Voici les deux formats possibles d'une option :

Cas 1: Une option codée sur un seul octet.
Cas 2: Un octet codant le type d'option, un octet donnant la taille de l'option, les octets de données d'option.
La taille de l'option compte tous les octets de l'option y compris le type, son propre octet et tous les octets de donnée d'option.

L'octet de type d'option est composé de trois champs de bits :

1 bit	indicateur de recopie
2 bits	classe d'option
5 bits	numéro d'option.

III. UDP : USER DATAGRAM PROTOCOL

Introduction

Le protocole User Datagram Protocol (UDP) est défini dans le but de fournir une communication par paquet unique entre deux processus dans un environnement réseau étendu. Ce protocole suppose l'utilisation du protocole IP comme support de base à la communication.

Ce protocole définit une procédure permettant à une application d'envoyer un message court à une autre application, selon un mécanisme minimaliste. Ce protocole est transactionnel, et ne garantit ni la délivrance du message, ni son éventuelle duplication. Les applications nécessitant une transmission fiabilisée et ordonnée d'un flux de données implémenteront de préférence le protocole TCP (Transmission Control Protocol) .

Format :

0 32 bits

Port source	Port destinataire
Longueur	Checksum
Données	

En-tête UDP

Champs

Le Port Source est un champ optionnel. Lorsqu'il est significatif, il indique le numéro de port du processus émetteur, et l'on supposera, en l'absence d'informations complémentaires, que toute réponse devra y être dirigée. S'il n'est pas utilisé, ce champ conservera une valeur 0.

Le Port Destinataire a une signification dans le cadre d'adresses Internet particulières.

La Longueur compte le nombre d'octets dans le datagramme entier y compris le présent en-tête. (Et par conséquent la longueur minimale mentionnée dans ce champ vaut huit, si le datagramme ne transporte aucune donnée).

Le Checksum se calcule en prenant le complément à un de la somme sur 16 bits des compléments à un calculé sur un pseudo en-tête constitué de l'information typique d'une en-tête IP, l'en-tête UDP elle-même, et les données, le tout additionné d'un octet nul éventuel afin que le nombre total d'octets soit pair.

La pré en-tête ajoutée avant l'en-tête UDP contient l'adresse IP source, l'adresse IP destinataire, le code de protocole, et la longueur du segment UDP. Cette informationpermet d'augmenter l'immunité du réseau aux erreurs de routage de datagrammes. La procédure de calcul du Checksum est la même que pour TCP.

Si le calcul du checksum vaut zéro, il sera transmis tous ses bits à un (le complément à un). UN Checksum transmis avec une valeur zéro a effectivement une signification particulière. Dans ce cas, le segment indique qu'aucun Checksum n'a été calculé (pour des besoins de mise au point ou pour des protocoles de niveaux supérieurs qui rendent cette vérification inutile).

Interface Utilisateur

L'interface utilisateur doit permettre l'ouverture de nouveaux ports de réception, la réception des données et leur transmission ainsi que celle de l'adresse source à l'application sur le port de réception mis en place, et doit mettre en place une commande permettant l'émission d'un datagramme, par laquelle seront spécifiés les données, l'adresse et ports source et destination à utiliser.

Interface IP

Le module UDP doit extraire les adresses source et destination de l'en-tête IP, et vérifier le numéro de protocole. Une interface UDP/IP plausible pourrait retourner le datagrammeentier y compris l'en-tête Internet en réponse du datagramme reçu. Une interface devra pour cela permettre à UDP de passer un datagramme Internet complet avec une en-tête IP à la couche IP elle même pour émission. IP n'aura plus qu'à vérifier la cohérence des champs d'en-tête IP préparés par UDP et calculer le Checksum.

Applications du Protocole

Ce protocole sera utilisé principalement pour les communications avec les serveurs de noms de domaines , et dans les transactions utilisant le protocole Trivial File Transfer.

Numéro de protocole

Ce protocole porte le numéro 17 (21 en octal) lorsqu'il est transporté par le Protocole Internet. D'autres numéros de protocoles pour d'autres couches support sont données dans la référence .

IV. ICMP : INTERNET CONTROL MESSAGE PROTOCOL

Introduction

Le Protocole Internet (IP) est utilisé pour la transmission de datagrammes de hôte à hôte à l'intérieur d'un système de réseaux interconnectés appelé Catenet .
Les appareils raccordant les réseaux entre eux sont appelés des Routeurs. Ces routeurs communiquent entre eux en utilisant le protocole Routeur à Routeur (GGP) afin d'échanger des informations de contrôle et de gestion du réseau. Occasionnellement, un routeur ou un hôte destinataire peut avoir à communiquer vers l'émetteur du datagramme, par exemple, pour signaler une erreur de traitement du datagramme. C'est dans cette perspective qu'a été mis en place le protocole Internet Control Message Protocol (ICMP). Il s'appuie sur le support de base fourni par IP comme s'il s'agissait d'un protocole d'une couche supérieure. ICMP n'en reste pas moins une partie intégrante du protocole IP, et doit de ce fait être implémenté dans chaque module IP.

Les messages ICMP sont envoyés dans diverses situations: par exemple, lorsqu'un datagramme ne peut pas atteindre sa destination, lorsque le routeur manque de réserve de mémoire pour retransmettre correctement le datagramme, ou lorsque le routeur décode de viser l'hôte destinataire via une route alternative pour optimiser le trafic.

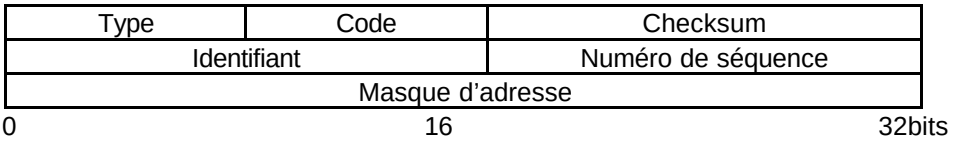
Le protocole Internet n'est pas, dans sa définition, absolument fiable. Le but de ces messages de contrôle est de pouvoir signaler l'apparition d'un cas d'erreur dans l'environnement IP, pas de rendre IP fiable. Aucune garantie que le datagramme soit acheminé ni qu'un message de contrôle soit retourné, de peut être donnée.
Certains datagrammes pourront se perdre dans le réseau sans qu'aucun message de contrôle ne le signale. Les protocoles de niveau supérieur s'appuyant sur une couche IP devront implémenter leurs propres mécanismes de contrôle d'erreur et de retransmission si leur objet nécessite un circuit de communication sécurisé.

Les messages ICMP reportent principalement des erreurs concernant le traitement d'un datagramme dans un module IP. Pour éviter de ne pas entrer dans un cercle vicieux de réémission de message de contrôle en réponse à un autre message de contrôle et ce sans fin, aucun message ICMP ne sera réémis en réponse à un message ICMP. De même les messages ICMP ne seront transmis qu'en réponse à un traitement erroné du fragment zéro dans le cas d'un datagramme fragmenté. (Le fragment zéro est celui dont l'offset vaut zéro).

Formats de message

Les messages ICMP sont émis en utilisant l'en-tête IP de base. Le premier octet de la section de données du datagramme est le champ de type ICMP; Sa valeur détermine le format du reste des données dans le datagramme ICMP.

En-tête ICMP



Résumé des types de Message (champ Type)

0 Réponse Echo
3 Destination non accessible
4 Contrôle de flux
5 Redirection
8 Echo
11 Durée de vie écoulée
12 Erreur de Paramètre
13 Marqueur temporelle
14 Réponse à marqueur temporel
15 Demande d'information
Réponse à demande d'information

Champ Identifiant : un identifiant pour aider à qualifier les réponses/requêtes (souvent à zéro)

Champ Numéro de séquence : un numéro de séquence pour aider à qualifier les réponses/requêtes (à zéro)

V. SNMP : Simple Network Management Protocol

1.Introduction

Le protocole SNMP est le langage que les agents et les stations de gestion (managers) utilisent pour communiquer. C'est un protocole de type question/réponse asynchrone. Ce protocole est situé au niveau application du modèle OSI, c'est lui qui définit la structure formelle des communications. SNMP est encapsulé dans des trames UDP.
La MIB (Management Information Base) regroupe l'ensemble des variables relatives aux matériels et aux logiciels supportés par le réseau, et définit les objets de gestion dans l'environnement TCP/IP.
La SMI (Structure of Management Information), définit comment sont représentées , dans la MIB, les informations relatives aux objets de gestion et comment sont obtenues ces informations.

Les stations interrogent donc les agents pour observer leur fonctionnement et leur envoient des commandes pour leur faire exécuter certaines taches. Les agents renvoient les informations requises aux stations de gestion. Certains événements du réseau, tels que des erreurs de transmission, peuvent déclencher des alarmes envoyées aux stations de gestion. Cependant, l'envoi de messages de façon spontanée de l'agent vers le manager est limité. Les managers effectuent une interrogation périodique des agents de manière à vérifier leur état. La structure des paquets est définie en utilisant la syntaxe ASN1 (Abstract Syntax Notation).

SNMP a l'avantage d'être simple, cependant il a des capacités très limités au niveau sécurité, principalement pour l'authentification. Tous les systèmes SNMP doivent également supporter les protocoles DUPER et IP pour transporter les données entre les agents et les stations de gestion.

2.Spécifications

Le format de la trame SNMP est décrit ci-dessous :

Version	Communauté	PDU

Champs

Version : numéro de version SNMP. Le manager et l'agent doivent utiliser le même numéro.

Communauté : ce champ sert à identifier auprès du manager l'agent avant de lui accorder un accès.

PDU : il y a 5 types de PDU : GetRequest, GetNextRequest, GetResponse, SetRequest, et TRAP.
Une description de ces PDU est données ensuite.

Un premier format est utilisé pour les PDU du genre GET, ou SET :

Type de PDU	ID de requête	Statut d'erreur	Index d'erreur	Obj 1, val 1

Champs

Type de PDU : 0 : GetRequest
1 : GetNextRequest
2 : GetResponse
3 : SetRequest

ID de requête : champ qui coordonne la requête du manager et la réponse de l'agent.

Statut d'erreur : entier qui indique une opération normale (cas 0) ou bien une erreur(cas $\neq 0$)

Index d'erreur : identifie les entées avec la liste des variables qui ont causé l'erreur.

Obj/Val : association du nom de la variable à transmettre avec sa valeur.

Un second format est utilisé pour la TRAP PDU :

Type de PDU	Entreprise	Adresse Agent	Type Generique	Type Specifique	Timestamp	Obj 1, val 1

Champs

Type de PDU : dans ce cas toujours égal à 4.

Entreprise : identifie l'entreprise de management qui a défini la Trap .

Adresse Agent : adresse IP de l'agent.

Type Générique : décrit quel type de problème est survenu. (7 valeurs sont possibles).

Type Spécifique : est utilisé afin d'identifier une TRAP spécifique à une entreprise.

Timestamp : contient la valeur de l'objet sysUptime représentant le temps écoulé depuis la dernière initialisation..

Obj/Val : association du nom de la variable à transmettre avec sa valeur.

VI. ARP : ADDRESS RESOLUTION PROTOCOL

1.Introduction

Les adresses IP sont attribuées indépendamment des adresses matérielles des machines. Pour envoyer un datagramme dans l'internet, le logiciel réseau doit convertir l'adresse IP en une adresse physique qui est utilisée pour transmettre la trame. Si l'adresse physique est un entier court, elle peut être facilement modifiée pour lui faire correspondre l'adresse machine IP. Sinon, la traduction doit être effectuée dynamiquement.

C'est le protocole ARP qui effectue cette traduction en s'appuyant sur le réseau physique. ARP permet aux machines de résoudre les adresses sans utiliser de table statique. Une machine utilise ARP pour déterminer l'adresse physique destinataire en diffusant (broadcast), sur le sous réseau, une requête ARP qui contient l'adresse IP à traduire. La machine possédant l'adresse IP concernée répond en renvoyant son adresse physique. Pour redre ARP plus performant, chaque machine tient à jour, en mémoire, une table des adresses résolues et réduit ainsi le nombre d'émissions en mode broadcast.

2.Spécifications

La structure d'une trame ARP est définie ci-dessous :

0 16 32 bits

Type Hardware		Type de protocole
Hlen	Plen	Opération
Adresse hardware de l'expéditeur		
Adresse protocole de l'expéditeur		
Adresse hardware du destinataire		
Adresse protocole du destinataire		

Champs

Type Hardware : spécifie le type de l'interface hardware

Type de protocole : spécifie le type du protocole de haut niveau émis par l'expéditeur

Hlen : longueur de l'adresse hardware

Plen : **longueur de l'adresse de haut niveau**

Opération : type de l'opération effectuée :
Requête ARP
Réponse ARP
Requête RARP
Réponse RARP
Requête RARP dynamique
Réponse RARP dynamique
Erreur RARP dynamique
Requête InARP
Réponse InARP

Adresse hardware de l'expéditeur : explicite

Adresse protocole de l'expéditeur : explicite

Adresse hardware du destinataire : explicite

Adresse protocole du destinataire : explicite

VII. RIP2 : ROUTING INFORMATION PROTOCOL 2

1.Introduction

RIP2 est utilisé pour échanger des informations de routage. Il dérive d'un premier protocole développé par Xerox (RIP). Chaque machine qui utilise un protocole RIP2 a un processus qui envoie et reçoit des datagrammes transportés par de l'UDP port numéro 520.
La structure des trames RIP2 est décrite ci-dessous :

0 16 32 bits

Commande	Version	Inutilisé
Identifiant de la famille d'adresses		Route tag
Adresse IP		
Masque de sous réseau		
Saut suivant		
Métrique		

Commande : utilisé pour définir le sujet du datagramme
Requête
Réponse
Réservé (Utilisé par Sun microsystems)

Version : numéro de la version RIP.

Identifiant de la famille d'adresses : indique quel type d'adresse est utilisé cela car RIP2 peut transporter d'autres informations de routage.

Route tag : (utilisé par RIP2 ; 0 pour RIP) attribue une route qui doit être préservée par une route. Ce champ permet de séparer les routes RIP internes des routes externes qui ont pu être importée d'un EGP ou d'un autre IGP.

Adresse IP : adresse IP de la destination.

Masque de sous-réseau : (utilisé par RIP2; 0 pour RIP) masque du sous réseau destination.

Saut suivant: adresse IP à laquelle les paquets devront être envoyé au prochain saut.

Métrique : représente le coût total de la source à la destination (en nombre de sauts).

LE PROTOCOLE X25

Introduction

Le protocole X25 définit l'interface entre un ETTD (Equipement Terminal de Traitement des Données) et un ETCD (Equipement Terminal de Circuit de Données). Il a été adopté par le CCITT en septembre 1976. On entend souvent par X25 l'ensemble des protocoles liés à X25 et qui couvre les couches 1 à 3 du modèle OSI. Pourtant, le terme X25 désigne uniquement le niveau 3 ou niveau paquet transporté entre les champs d'information des trames LAPB.

I. LAPB

Le protocole LAPB est le protocole de niveau 2 qui transporte les paquets X25. Le format standard d'une trame LAPB est le suivant:

Flag	Champ adresse	Champ de contrôle	Données	FCS	Flag

Flag: Toujours 0x7E

Address Field: Ce champ n'a aucune raison d'être quand on travaille de point à point. Cet octet est réservé à plusieurs utilisations. Il sert à séparer les commandes des réponses et peut seulement prendre les valeur 0x01 et 0x03. 01 désigne une commande de l'ETTD à l'ETCD et 03 contient une réponse de l'ETCD à l'ETTD.

Champ de contrôle: Identifie le type de trame. En plus, il inclut la séquence de nombre, les fonctions de contrôles et le traquage des erreurs en fonction du type de trame.

FCS: Frame Check Sequence.

Type de trame:
Trame de supervision:
RR : Prêt à recevoir.
REJ : Demande de retransmission.
RNR : Pas prêt à recevoir.

Trames non séquentielles:
DISC : Demande de déconnexion.
UA : Trame d'acquitement.
DM : Réponse à DISC, mode déconnexion.
FRMR: Rejet de trame.
SABM: Mode asynchrone, pas de maître et d'esclave.

Trame d'information:
INFO

II. X25

La structure du paquet de données X25 est la suivante:

8	7	6	5	4	3	2	1
Q	D	0	1	Numéro de groupe de voie logique			
Numéro de voie logique							
P(R)		M		P(S)		O	
Données							

GFI: Identifiant de format général. Q indique un paquet X25 (0) ou X29 (1). D indique un acquittement local (0 : ETCD) ou distant (1 : ETTD). Les bits 01 indiquent que les numéros de trames vont de 0 à 7. Le format de trame où ils indiquent 10 montre que l'on numérote les trames de 0 à 127 (10). Cela permet d'envoyer beaucoup de trame avant d'acquitter ce qui est intéressant pour les réseaux lents tels que les réseaux sattelites.

Type de paquet:
P(R) : Nombre des paquets reçus.
P(S) : Nombre de paquets envoyés.
M : Seulement dans les paquets de données. Ce champ indique, lorsqu'il est à 1, que le paquet fait partie d'un ensemble de paquets à traiter comme un tout.

Les paquets peuvent être de différents types:
CALL ACC : Appel accepté.
CALL REQ : Demande d'appel.
CLR CNF : Confirmation d'effacement.
CLR REQ : Demande d'effacement.
DATA : Paquet de données
DIAG : Diagnostique.
INF CNF : Confirmation d'interruption.
INT REQ : Demande d'interruption.
REJ : Rejet.
RES CNF : Confirmation de remise à zéro.
RES REQ : Demande de remise à zéro.
RNR : Non prêt à recevoir.
RR : Prêt à recevoir.
RSTR CNF : Confirmation pour recommencer.
RSTR REQ : Demande qu'on recommence.
REG REQ : Demande de registration.
REG CNF : Confirmation de registration.

III. HDLC

Flag	Adresse	Champ de contôle	Données	FCS	Flag

Flag: Toujours 0x7E

Champ de contrôle: Indique le type de trame auquel on a affaire. Les différents types de trames comprennent les même types de trames que pour le protocole LAPB plus d'autres énumérées ci-après:

Trame de supervision:
SREJ : Demande de retransmission d'une trame.

Trames non séquentielles:
SARM : Mode de réponse asynchrone. Demi-relation maître/esclave.
REST : Remise à zéro du nombre de trame.
CMDR : Commande rejetée.
SNRM : Mode de réponse normal. Relation maître/esclave.
RD : Requete déconnectée.
RIM : Deuxième demande d'initialisation après déconnection.
SIM : Mode d'initialisation.
UP : Election non séquentielle.
UI : Information non séquentielle.
XID : Commande d'échange d'identification.

RNIS : RESEAU NUMERIQUE A INTEGRATION DE SERVICES

1.Architecture

Les services de transmission de données se sont développés, depuis le début des années 70, sur le principe des réseaux spécialisés : à un usage correspondait un réseau spécifique. L'utilisateur qui avait besoin de communiquer avec chacun de ces réseaux était donc obligé d'avoir autant de raccordements que de réseaux ou d'applications à atteindre.

Cette multitude de raccordements différents et indépendants n'était optimale ni du point de vue de l'utilisateur ni point de vue de l'exploitant Télécom ; de cette constattation est né le concept d'intégration de services.
Ce réseau s'appuie sur le concept RNIS (Réseau Numérique à Intégration de Services) ou ISDN (Integrated Services Digital Network). Le RNIS propose l'intégration des supports et des services et, pour cela, il s'appuie sur la numérisation et se développe au sein d'une structure puissante de normes internationales.

Le RNIS est une évolution du réseau téléphonique actuel. Il propose la continuité numérique de bout en bout. Ce n'est pas un réseau supplémentaire entrant en concurrence avec les réseaux existants comme le téléphonique traditionnel, les réseaux X.25 ou les liaisons spécialisées. C'est plutôt un accès universel à ces réseaux ou plus exactement à ces services supports.

En jouant sur son sigle , le RNIS apparaît comme un moyen de communication rapide, normalisé, intelligent et souple :

rapide, car l'accès de base à 144 Kbit/s comporte 2 voies à 64 Kbit/s et une voie à 16 Kbit/s (2B+D). Les canaux B permettent, par exemple, de téléphoner tout en envoyant une télécopie rapide. Le canal D, pour sa part, convoie les signaux servant à l'établissement de la communication et toutes les informations de service ; il peut aussi transporter des informations à bas débit. Il existe des accès primaires qui comportent 30 canaux B et un canal D.
Normalisé, car tous les éléments d'accès au RNIS sont spécifiés par des normes internationales : même canal de base, même canal D, même cablâge et même prise (RJ 45) servent pour tous.
Intelligent, car les centraux sont capables de gérer une signalisation bien plus riche que celle du téléphone classique. Cette possibilité offre un grand nombre de services complémentaires comme, par exemple, l'identification de l'appelant ou la possibilité de transfert d'appel. Par ailleurs, il existe un contact permanent entre l'abonné et le réseau ; par exemple, si un abonné occupe ses 2 canaux B avec une communication téléphonique et un transfert de données, le réseau pourra, grâce au canal D, avertir l'utilisateur qu'un autre correspondant cherche à le joindre.
Souple et simple, car le RNIS a la vocation d'héberger la grande majorité des services de communication et fait un pas vers la transparence des réseaux avec son accès universel aux services de télécommunication.

2.Spécifications

Le LAPD (Link Access Protocol – channel D) is un protocole de niveau 2 qui travaille avec l'Asynchronous Balanced Mode (ABM). Ce mode est complètement équilibré (ni maitre, ni esclave). Chaque station peut intialiser, superviser et envoyer des trame à tout moment.

Le format de la rame est décrit ci-dessous :

Flag	Adresse	Contrôle	Information	FCS	Flag

Champs :
Flag : Sa valeur est toujours (0x7E). De sorte à ce que le flag ne soit pas dupliqué dans la trame on utilise la technique du Bit Stuffing.

Adresse : Les 2 premiers octets de la trame après le champ flag sont les champs d'adresse. Le format de Adresse est le suivant :

1	2	3	4	5	6	7	8
SAPI						C/R	EA1
TEI							EA2

EA1 : premier bit d'extension d'adresse qui est toujours 0

C/R : bit Commande/Réponse. Les trames qui proviennent d'un utilisateur avec ce bit à 0 sont des trames de commande

SAPI : Service Access Point Identifier.
Les valeurs sont :
procédure appel-contrôle
mode communication de paquets utilisant la procédure d'appel-contrôle I.451
communication de paquets X.25 niveau 3
manager de procédures niveau 2

Contrôle : identifie le type de trame avec un contrôle sur la trame

FCS : Frame Check Sequence.

STRUCTURE DE LA TRAME RNIS

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Discriminateur protocole							
0	0	0	0	Longueur valeur appel ref.			
Flag	Valeur appel ref.						
0	Type message						
Autres éléments requis							

Champs :

Discriminateur protocole : protocole utilisé

Longueur valeur appel référence : détermine la longueur du champ suivant. La référence d'appel peut être d'une longueur de 1 ou 2 octets qui dépend de la taille de la valeur codée.

Flag : mis à 0 pour les messages émis par le parti qui à alloué la valeur de l'appel de référence ; autrement mis à 1.

Valeur appel ref : une valeur arbitraire est allouée pour la durée de la session, qui identifie l'appel entre la machine maintenant l'appel et le switch RNIS

Type message : définit le premier sujet de la trame. Le type de message peut être de 1 ou 2 octets. Quand il y a plus d'un octet, le premier octet ne contient que des 0.

LES PROTOCOLES NOVELL

1.Introduction

L'ensemble des protocole Novell Netware a été grandement influencée par le design et l'implémentation de l'architecture de protocoles du Xerox Network System (XNS). Il fournit un support compréhensible par DOS, Windows, Macintosh, OS/2 et UNIX. De plus, Novell fournit un large support aux réseaux locaux et aux communications asynchrones de large zone. Novell comprend les protocoles suivants :
IPX – Internetwork Packet Exchange
RIPX – Routing Information Protocol
BCAST – Broadcast
DIAG – Diagnostic Responder
SER – Serialization
WDOG – Watchdog
SPX – Sequenced Packet Exchange
SAP – Service Advertising Protocol
NovelNetBios
BMP – Burst Mode Protocol
NCP – Netware Core Protocol
NDS – Netware Directory Services

I. IPX

1.Spécifications
IPX est l'implémentation Novell du Internet Datagram Protocol (IDP) développé par Xerox. IPX est un protocole datagramme sans connexion qui transmet des paquets à travers Internet et fournit aux stations Netware et aux serveurs de fichiers des services d'adressage et de routage inter réseaux.

La structure du paquet IPX est décrite ci-dessous :

0 15 bits

Checksum	
Longueur du paquet	
Contrôle transport	Type paquet
Réseau destination (4 octets)	
Nœud destination (6 octets)	
Socket destination	
Réseau source (4 octets)	
Nœud source (6 octets)	
Socket source	
Données (contient des info. RIP, SAP,SPX….)	

Champs

Checksum : mis à FFFFH

Longueur du paquet : longueur du datagramme IPX en octets

Control transport : utilisé par les routeurs Netware. Mis à zéro avant une transmission de paquet

Type de paquet : spécifie l'information contenue dans le paquet
Hello ou SAP
RIP
Paquet ECHO
Paquet erreur
Netware 386 ou SAP
Protocole de séquencement de paquets
16/31 Protocoles expérimentaux
Netware 286

Numéro de réseau : nombre sur 32 bits déterminé par l'administrateur réseau. 0 en local

Numéro de nœud : nombre sur 48 bits qui identifie l'adresse hardware LAN. Si ce nombre est FFFF FFFF FFFF FFFF c'est un broadcast.

Numéro socket : nombre sur 16 bit qui identifie la paquet de haut niveau

0451H	NCP
0452H	SAP
0453H	RIP
0455H	Netbios
0456H	diagnostique
0457H	Paquet de serialisatin (SER)

4000-6000H Sockets éphémères utilisées pour les communications des serveurs de fichiers et des réseaux

II. RIPX

1.Spécifications

RIPX est utilisé pour collecter, maintenir et échanger des informations de routage correctes entre les passerelles dans Internet. Il ne faut pas confondre ce protocole avec celui de la famille TCP/IP ! !

La description du paquet est donnée ci-dessous :

15

Opération
Numéro réseau
Nombre de sauts
Nombre de ticks
. . . . .

Champs :
Opération : Spécifie le type d'opération
Requête RIP
Réponse RIP

Numéro réseau : adresse sur 32 bits du réseau spécifié

Nombre de sauts : nombre de routeurs jusqu'au réseau spécifié

Nombre de ticks : mesure de temps nécessaire pour atteindre le réseau spécifié (18,21 ticks/seconde)

III. BCAST

Spécifications
BCAST sert à diffuser les annonces du réseau informant l'utilisateur qu'il a bien reçu un message.

La description de la trame est expliquée ci-dessous :

0 15 bits

Numéro de connexion	Signature

Champs :

Numéro de connexion : communiqué à la station durant le processus de login

Signature : la valeur est 0x21 (caractère ASCII) qui signifie *broadcast message waiting.*

IV. DIAG

DIAG est très utilisé pour analyser les LAN Netware. DIAG peut être utilisé pour tester les connexions et les configurations.

La structure des paquets est décrite ci-dessous :

Compteur d'exclusion d'adresse (1 octet)
Exclusion d'adresse 0 (6 octets)
. . . .
Exclusion d'adresse 79 (6 octets)

Paquet de requête DIAG

Champs :
Compteur d'exclusion d'adresse : le nombre de stations à qui on demande de ne pas répondre. 0 dans ce champ signifie que l'on demande à toutes les stations de répondre.

0 16 bits

Major version	Minor version
SPX diagnostic socket	
Component count	
Component Type 0 (longueur de la variable)	

Paquet de réponse DIAG

Champs :

Major/Minor version : version du diagnostique installé dans la station qui répond

SPX diagnostic socket : le numéro de socket auquel tous les diagnostiques réponses SPX peuvent être adressés

Component count : nombre de composants trouvés dans le paquet réponse

Component Type : contient des informations à propos d'un des composants ou processus actif au nœud répondant.

Simple :

0= IPX/SPX
1= drivers de routeurs
2= drivers de LANs
3= Shells
4= VAPs

Etendu :

5= Routeur
6= Serveur de fichiers/routeur
7= IPX/SPX non dédiés

Chaque champ étendu est suivi de champs additionnels :

Nombre de réseaux locaux (1 octet)

Champ spécifique :

Nombre de réseaux locaux : nombre de réseaux locaux avec lesquels le composant peut communiquer.

Pour chaque réseau local il y aura en plus :

Type du réseau local (1 octet)
Adresse réseau (4 octets)
Adresse nœud (6 octets)

Type du réseau local : contient un nombre indiquant le type de réseau local avec lequel va communiquer le composant

Adresse réseau : contient l'adresse sur 4 bits du réseau définit dans le cham précédent

Adresse nœud : contient l'adresse sur 6 bits du nœud qui accompagne l'adresse réseau

V. SER

Pour assurer qu'une simple version de Netware n'est pas en train d'être chargée sur différents serveurs, l'OS diffuse des paquets appelés paquets de sérialisation pour déterminer si il y a multiple copies du même OS sur le réseau.
Ces paquets ne contiennent qu'un seul champ de 6 octets appelé champ de donnée de sérialisation.

VI. WDOG

Spécifications
Le WDOG fournit les validations de connexion aux stations sur le système Netware et rend compte à l'OS lorsque une connexion doit être fermée pour cause de trop longue période sans communications.

La description de la trame est donnée ci-dessous :

Numéro de connexion	Signature

16bits

Champs

Numéro de connexion : donné à la station durant le login

Signature : contient un caractère 0x3F ou 0x59

VII. SPX

SPx est la version Novell du Sequenced Packet Protocol de Xerox (SPP). C'est un protocol situé au niveau de la couche liaison et il permet une distribution des paquet à des applications tierces.
La description du paquet SPX est faite ci-dessous :

16bits

Flag contrôle connexion	Type flux données
ID source connexion	
ID destinataire connexion	
Numéro de séquence	
Numéro acquittement	
Numéro allocation	
0-534 octets de données	

Champs

- **Flag contrôle connexion :** 4 flags qui contrôlent le fux bidirectionnel à travers une connexion SPX. Lz valeur est 1 pour un bit mis, 0 sinon.
  - Bit 4 Eom :end of message
  - Bit 5 Att : pas utilisé par SPX
  - Bit 6 Ack : pas utilisé
  - Bit 7 Sys : contrôle de transport
- **Type flux données :** spécifie la donnée
  0-253 ignoré par SPX
  Fin de connexion
  Ack de fin de connexion
- **ID source connexion :** nombre sur 16 bits assigné par SPX pour identifier la connexion
- **ID destination connexion :** le nombre de référence utilisé pour identifier la destination du transport
- **Numéro de séquence :** nombre de 16 bits contrôlé par SPX qui indique le nombre de paquets transmis
- **Numéro d'acquittement :** nombre sur 16 bits qui indique le prochain paquet
- **Numéro d'allocation :** nombre sur 16 bits qui indique le nombre de paquets envoyés mais pas encore acquittés.

VIII. SAP

Spécifications
Avant qu'un client ne puisse communiquer avec un serveur, il doit savoir quels serveur sont disponibles sur le réseau. Cette information est rendue disponible grâce au Novell's Service Advertsing Protocol (SAP). Le service SAP diffuse l'information sur la liste des serveurs connu à travers tout le réseau.Ces serveurs peuvent comprendre des serveurs de fichiers, des serveurs d'impression, des serveurs d'accès Netware, et des serveurs distants.

Le format de la trame SAP de réponse est décrite ci-dessous :

Opération (2octets)
Type service (2 octets)
Nom serveur (48 octets)
Adresse réseau (4 octets)
Adresse nœud (6 octets)
Adresse socket (2 octets)

Sauts (2 octets)
….

Le paquet SAP peut contenir l'information de 7 serveurs.
Champs

Opération : spécifie l'opération que le paquet va effectuer :
1 Requête de service général
2 Réponse de service général
3 Requête du service le plus proche
4 Réponse du service le plus proche

Type service : spécifie le service exécuté :
01H Utilisateur
04H Service de fichiers
07H service d'impression
21H passerelle NAS SAN
23H NACS
27H passerelle TCP/IP
98H serveur d'accès Netware
107H Netware386 STOREXP Spec.
137H Netware 386 queue d'impression

H signifie hexadécimal.

- **Nom serveur :** contient sur 48 octets le nom du serveur
- **Adresse réseau :** numéro du serveur de réseau sur 32 bits
- **Adresse nœud :** numéro du serveur de nœuds sur48 bits
- **Adresse socket :** numéro de serveur de socket sur 16 bits
- **Sauts :** nombre de routeurs par lesquels sont passés les paquets pour atteindre le réseau spécifié.

Opération (2 octets)
Type service (2 octets)

Format du paquet de requête SAP

IX. NovelNetBios

Spécifications

C'est un protocole propriétaire développé par Novell basé sur NetBios.
Le champ type de flot de données est constitué d'un octet. Les autres champ sont de taille variable.
Exemples pour l'octet type de flot de données :
Trouve nom
Nom reconnu
Vérifie nom
Nom utilisé
De-register nom
Session données
Session fin
Session fin ack
Statut requête
Statut réponse
Datagram direct

X. BMP (Burst)

Spécifications
Le BMP est en fait un type de paquet NCP (Request type=7777H). BMP a été créé afin de permettre de multiple réponses pour une seule requête et donc transférer jusqu'à 64 ko de données pour une seule requête en lecture de fichiers.

Le format de la trame BMP est donné ci-desous :

	16	24	32 bits
Type requête	Flag type flot	Type flot	
ID source connexion			
ID destinataire connexion			
Numéro de séquence du paquet			
temps délais			
Numéro de séquence burst	Numéro d'ACK de séquence		
Longueur totale du burst			
Offset total du burst			
Longueur du paquet	Nombre des entrées de liste		
Fragment manquant de la liste			
Code fonction			
Handle de fichier			
Offset de début			
Octets à écrire			

Champs

- **Type requête :** Identique à type requ^te dans NCP et toujours positionné à 7777H (mode paquets burst)
- Flags type flot : **flags disponibles**
- **Type flot :** contrôle du mode Burst
- **ID source connexion :** numéro d'ID assigné à la station source

ID detinataire connexion : numéro d'ID assigné à la station destination

Numéro de séquence du paquet : **utilisé par la station et les serveurs de fichiers pour identifier les paquets envoyés et reçus**

- **Temps de délais :** délais entre 2 paquets
- **Numéro de séquence burst :** numéro de la séquence en cours de transmission
- **Numéro de séquence ACK :** numéro de la prochaine séquence burst acceptée
- **Longueur totale burst :** loongueur de la burst transmise (en octets)
- **Offset burst :** position des donnés dans le burst
- **Longueur paquet :** longueur des données burst (en octets)
- **Nombre d'entrées de liste :** nombre d'éléments dans la liste des fragments manquant

Liste des fragments manquants : fragment de données encore non reçues

Code fonction : fontion lire ou écrire

Offset début : Offset du début écriture (/lecture)

XI. NCP

Spécifications
Le Novell Netware Core Rotocol (NCP) s'occupe de l'accès ressources du serveur primaire Netware. Il fait des appels de procédures au NFSP (Netware File Sharing Protocol) .

Le format du paquet NCP est décrit ci-dessous :
Le type requête est sur 2 octets ; tous les autres sont sur un octet.

Type requête
Numéro séquence
Numéro de connexion bas
Numéro de task
Numéro de connexion haut
Code requête
Données

En-tête du paquet requête

Champ

Type requête : identifie le type de paquet

1111H	Requête d'allocation de slot
2222H	Requête de serveur de fichiers
3333H	Réponse de serveur de fichier
5555H	Requête de désallocation de slot
7777H	Mode BMP
9999H	Ack positif

H signifie hexadécimal.

Numéro de séquence : numéro utilisé par la station et les serveurs de fichiers pour identifier les paquets qui sont envoyés et reçus

Numéro de connexion bas : ID de connexion bas associé à la station.

Numéro de task : identifie le système d'exploitation, DOS…

Numéro de connexion haut : ID de connexion haut associé à la station. Utilisé seulement sur la version 1000-user de Netware, sur les autres versions il est à zéro

Code requête : identifie le code spécifique de la requête

A présent voici le paquet de réponse NCP qui est le même que celui de requête NCP excepté les deux octets suivant l'octet Numéro de connexion haut. Ces deux octets sont définis ci-dessous :

Code completion
Statut connexion

Code completion : ce code indique si la requête du client a abouti ou pas. Une valeur de 0 indique une réussite.

Statut connexion : le 4eme bit sera mis à un si le serveur doit être arrêté

IPV6

POURQUOI UNE NOUVELLE VERSION D'IP ?

IPng est une nouvelle version d' IP qui s'inscrit comme l'évolution naturelle et normale du protocole IP en place IPv4. Protocole IPv4 qui, tout en ayant permis l'énorme croissance de l'Internet, souffre de plusieurs faiblesses. IPng est conçu pour fonctionner aussi bien sur des réseaux à très hauts débits comme ATM que sur des réseaux à faible bande passante tels que les réseaux sans fils.
Nous allons voir que le développement d' IPng, malgré les limitesd'IPv4, correspond aussi à une réponse à des besoins croissants sur des nouveaux marchés en pleine expansion ainsi qu'à un effort de modernisation et d'évolution de la technologie proposée par l'Internet.

Les faiblesses du protocole IP actuel (IPv4)

Dès sa création, IP a suscité de nombreuses discussions notamment sur la conception de l'en-tête [RFC 791].
Le problème le plus connu concerne l'espace d'adressage. Les adresses IP sont actuellement stockées sur 32 bits ce qui permet environ plus de quatre milliards d'adresses, taille largement suffisante à l'origine lorsque le modèle dominant était celui d'un ordinateur par campus ou centre de recherche. Aujourd'hui, l'informatique industrielle et commerciale ainsi que celle des particuliers rendent ce nombre trop faible, d'autant que de nombreuses adresses sont "gaspillée" par le mécanisme d'allocation hiérarchique. En outre, la généralisation des machines connectées en réseau("toasternet problem" ou "paradigm shift") risque d'aggraver ce problème.
La description de l'en-tête IPv4 nous permet de mieux appréhender les limites du protocole actuel.

(chaque ligne du tableau représente 32 bits)
IHL : Internet Header Length,
Fragment Offset : la place du fragment,
Padding : le bourrage.
Un autre problème est celui posé par l'explosion de la taille des tables de routage dans l'Internet. Le routage, dans un très grand réseau, doit être hiérarchique avec une profondeur d'autant plus importante que le réseau est grand. Le routage IP n'esthiérarchique qu'à trois niveaux : réseau, sous-réseau et machine. Les routeurs des grands réseaux d'interconnexion doivent posséder une entrée dans leurs tables pour tous les réseaux IP existants. Ce problème est particulièrement résolu par le "supernetting" ou CIDR (Classless Internet Domain Routing) [RFC 1519].
IPv4 ne permet pas d'indiquer de façon pratique le type de données transportées (TOS ou Type of Service dans IPv4)d'où, par conséquent, leur urgence ou le niveau de service souhaité. Ce besoin est particulièrement critique pour les applications "temps réel" comme la vidéo mais aussi celles plus classiques (par exemple, en donnant des priorités à tel ou tel trafic). Ce problème a été évoqué et clarifié mais reste peu mis en oeuvre [RFC 1349]. Il est symptomatique que les protocoles de routage les plus répandus ne tiennent pas vraiment compte du TOS dans le calcul des routes.
Enfin, IPv4 ne fournit pas de mécanismes de sécurité comme l'authentification des paquets, leur intégrité ou leur confidentialité. Il a toujours été considéré que ces techniques étaient de la responsabilité des applications elles-mêmes.
L'évolution des besoins des utilisateurs ainsi que l'apparition de nouveaux marchés ne cessent d'amplifier les "carences" du protocole IPv4 actuel.

DESCRIPTION DES PRINCIPALES CARACTÉRISTIQUES D'IPv6

IPv6 est la nouvelle version d' IP et représente une très forte évolution par rapport à IPv4. Les principales fonctionnalités d'IPv4 sont conservées dans IPv6 excepté certaines fonctions peu ou pas utilisées qui ont été supprimées ou rendues optionnelles. En outre, quelques priorités ont été ajoutées.
Il est possible de dégager huit grandes caractéristiques inclues dans IPv6.

Des possibilités étendues d'adressage et de routage

La taille de l'adresse IP augmente de 32 à 128 bits afin de supporter un plus grand nombre de noeuds adressables, davantage de niveaux d' adressage hiérarchique ainsi qu'une auto configuration plus simple des adresses.
Un mécanisme adaptable de diffusion ainsi qu'un nouveau type d' adresses en "cluster" sont définis dans IPv6.

Un format d'en-tête simplifié

Des champs du format de l'en-tête IPv4 ont été abandonnés ou rendus optionnels, ainsi l'en-tête IPv6 est simplifié et réduite à un traitement commun dans tous les routeurs ce qui diminue donc le coût de traitement dans ces routeurs.

Des possibilités d'extension des en-têtes et desoptions

Dans IPv6, les options sont rangées dans des en-têtes supplémentaires situés entre l'en-tête IPv6 et l'en-tête du paquet de transport (T-PDU, Transport Protocol Data Unit ou Unités de données du service de transport). La plupart des options dans les en-têtes IPv6 ne sont ni examinées, ni traitées par les routeurs intermédiaires. Contrairement àIPv4, les options IPv6 peuvent être de longueur arbitraire, il n' existe pas de taille limite.
Une des caractéristiques d'IPv6 est la possibilité de coder, dans les options, l'action qu'un routeur ou une station de travail doit réaliser si l'option est inconnue, ce qui permet l'ajout de fonctionnalités supplémentaires dans un réseau déjà opérationnel avec un minimum de perturbations.

Des possibilités d'authentification et de confidentialité

IPv6 intègre des extension permettant l'authentification des usagers et l'intégrité des données grâce à des outils de cryptographie.

Des possibilités d' autoconfiguration

IPv6 dispose de plusieurs formes d' autoconfiguration comme la configuration "plug and play" d'adresses de noeuds sur un réseau isolé grâce aux caractéristiques offertes par DHCP.

Des possibilités pour le "Source Route"

IPv6 intègre une fonction étendue de source routing grâce à SDRP (Source Demand Routing Protocol) afin d'étendre le routage à des routes interdomaine et intradomaine.

Une transition d'IPv4 à IPv6 simple et flexible

La transition d'IPv4 à IPv6 répond à quatre objectifs essentiels :
un besoin de modernisation,
un besoin de redéploiement,
un adressage facile,
une diminution du coût de démarrage.
Tous ces aspects seront revus ultérieurement lors de l'étude approfondie de la transition d'IPv4 à IPv6 (cf. XII).

Des possibilités de qualité de service

L'introduction de flux étiquetés (avec des priorités), les services de contraintes <<temps réel>> sont de nouveaux éléments rendant possible la qualité de service.
Après cette énumération succincte rapide des principales caractéristiques d'IPv6, il convient de reprendre un certain nombre de ces éléments afin d'apporter quelques éclaircissements sur les nouvelles possibilités d'IPv6.

FORMAT D'EN-TÊTE IPv6

Avant toute chose, donnons quelques définitions :

Noeud (node) : un module protocolaire qui implémente IPv6.
Router (routeur) : un noeud qui transmet des paquets IPv6 non explicitement adressés à lui-même.
Station (host) : tout noeud qui n'est pas un routeur.
Interface : un attachement d'un noeud à une liaison.
Adresse (address) : un identifiant de la couche IPng pour une interface ou un groupe d'interfaces.
Liaison (link) : un moyen de communication ou médium sur lequel les noeudspeuvent communiquer avec la couche liaison (la couche immédiatement sous IPv6).
Voisins (neighbors) : noeuds rattachés à la même liaison.
MTU de liaison (link MTU) l'unité de transmission maximale (Maximum Transmission Unit - MTU), c'est à dire la taille maximale du paquet en octets, qui peut être acheminé en un seul "morceau" sur une liaison (un paquet étant composé d'une en-ête IPv6 et de son "payload").
MTU de chemin (path MTU) : le plus petit MTU de liaison de toutes les liaisons que compose un chemin entre un noeud source et un noeud de destination.
Bien que plus longue que la version 4 d'IP, l'en-tête de IPng a été simplifiée. Un certain nombre de fonctions présentent dans l'en-tête d'IPv4 ont été soit disposées dans des en-têtes supplémentaires, soit abandonnées.

- Version (4 bits): ce champ défini le numéro de version du Protocole IP. IPng est la version 6.
- Flow Label (28 bits): ce champ, étiquette de flux, peut être utilisé par une station pour "marquer" certains paquets afin qu'ils suivent un routage (service) particulier dans un réseau, tel un service de qualité sans défaut, ou un service "temps-réel".
- Payload Length (16 bits): elle représente la longueur des données après l'en-tête IPv6 en octets. Pour étendre cette valeur à plus de 64 KOctets, la valeur est donnée dans une option noeud-par-noeud. Le champ longueur données est alors à 0.
- Next Header (8 bits): il identifie le type d'en-tête suivant immédiatement l'en-tête IPv6. Ce champ est le même que le champ protocole d'IPv4.
- Hop limit (8 bits): utilisé pour détruire les paquets qui pourraient rester dans le réseau à la suite de boucles dues aux tables de routage, ce champ est décrémenté de une unité à chaque noeud qui retransmet le paquet (équivalent au Time To Live d'IPv4 - durée de vie).
- Source Address (128 bits): champ adresse de l'émetteur du datagramme.
- Destination address (128 bits): champ adresse du destinataire du paquet. Il est possible que l'adresse ne soit pas celle du destinataire final si une option de routage est présente.

EN-TÊTES SUPPLÉMENTAIRES

Dans la nouvelle version 6 d' IP, des informations complémentaires sont codées dans des en-têtes qui doivent être placés dans le paquet entre l'en-tête IPv6 l'en-tête de la couche transport. Ily a un petit nombre d'extensions d'en-tête, chacun identifié par une valeur de Next Header distincte. Comme illustré

dans les exemples suivants, un paquet IPv6 peut comporter aucune, une ou plus d'en-têtes supplémentaires,

Mise à part une exception, les en-têtes supplémentaires ne sont nullement examinés ou manipulés par les noeuds atteints parle paquet le long de son chemin, jusqu'à ce que le paquet arrive aunoeud (ou à chaque groupe de noeuds dans le cas du multicast)identifié par le champ adresse destinataire de l'en-tête IPv6. Ace moment là, le premier en-tête supplémentaire, ou l'en-tête transport dans le cas d'absence d'en-tête supplémentaire, est traité. Le contenu de chaque en-tête déterminera s'il faut, ou pas, traiter l'en-tête suivant.
La seule exception est l'en-tête de l'option noeud-par-noeud, elle porte des informations qui doivent être examinées par les noeuds du réseau. Cette en-tête "Hop-by-Hop" Options, lorsqu'elle est présente, doit suivre immédiatement l'en-tête IPv6.
Chaque en-tête supplémentaire est d'une longueur d'un multiple de8 octets, afin de conserver un alignement de 8 octets pour les en-têtes suivants.

Ordre des en-têtes supplémentaires

Lorsqu'il y a plus d'un en-tête supplémentaire utilisé dans le même paquet, les en-têtes doivent apparaître dans l'ordre suivant :
IPv6 Header
Hop-by-Hop Options Header
Routing Header
Fragment Header
Authentication Header
End-to-End Options Header
Chaque type d'en-tête ne doit apparaître qu'une seule fois dans le paquet (excepter dans le cas d'une encapsulation IPv6 dans IPv6, où chaque en-tête IPv6 encapsulé doit être suivi par son propre en-tête supplémentaire).

Option noeud-par-nœud

L'en-tête d'options noeud-par-noeud comporte des informations analysées par les différents noeuds du chemin pris par le paquet. L'en-tête des options noeud-par-noeud est identifié par une valeur de Next Header égale à 0, et a le format suivant :

- Next Header (8 bits): identifie le type d'en-tête suivant immédiatement l'en-tête d'options noeud-par-noeud. Les valeurs sont identiques au champ de protocole de IPv4.
- Hdr Ext Len (8 bits): longueur de l'en-tête des options noeud-par-noeud en multiple de 8 octets, à l'exclusion des 8 premiers.
- Options: ce champ contient une ou plusieurs options codées en TLV(Type-Length-Value). Ce premier est de longueur variable, il est un multiple de 8 octets. Nous allons décrire ce champ dans le paragraphe suivant.

Options d'en-tête IPv6

Deux des en-têtes supplémentaires actuellement définis, celle des options noeud-par-noeud et celle des options bout-en-bout, doivent porter un nombre variable d'options codées TLV suivant le format:

- Option Type (8 bits): identifiant de la nature de l'option.

- Opt Data Len (8 bits): longueur du champ de données de cette option en octets.

- Option Data (longueur variable): données de l'option. Champ de longueur variable.

Les identifiants Option Type sont codés de manière à ce que les deux bits de poids fort provoque l'opération suivante si un noeud ne reconnaît pas le Option Type:

00 ne traite pas cette option et poursuit le traitement de l'en-tête 01 détruit le paquet
10 détruit le paquet et envoie à l'adresse source un message ICMP de non reconnaissance, et indique la faute en donnant le Option Type.
11 non défini

Dans le seul cas des options noeud-par-noeud, le troisième bit de poids fort de Option Type indique si les données de cette option doivent être soumises au calcul d'assurance intégrité lorsque l'en-tête d'authentification est présente. Les données de l'option modifiées en cours de routage sont exclues de ce calcul.

inclus un calcul d'intégrité
exclus du calcul d'intégrité

Les champs Option Data des options de bout-en-bout ne changent jamais en route et donc, sont toujours inclus dans le calcul d'intégrité.

En-tête de routage

L'en-tête de routage est utilisé par une source pour établir une table de noeud(s) intermédiaire(s) (ou ensemble de groupes) que doit emprunter le paquet pour arriver à destination. Cette forme particulière d'en-tête de routage est conçue pour supporter le "protocole de routage à la demande de la source" (Source Demand Routing Protocol, SDRP) [Estrin94b].

Next Header: identifie le type d'en-tête suivant immédiatement l'en-tête de routage. Les valeurs sont identiques au champ de protocole de IPv4.

Routing Type: indique le type de routage supporter par cette en-tête. La valeur est 1.

MRE (Must Report Errors) flag: si ce bit est à 1 et qu'un routeur ne puisse émettre, conformément à la liste Source Route, le paquet (avec un routage incomplet), le routeur génère un message d'erreur ICMP. Dans le cas où le bit MRE est à 0, le routeur ne génère pas de message d'erreur ICMP.

F (Failure of Source Route Behavior) flag (1 bit): si ce bit est positionné à 1, il indique que si un routeur ne peut acheminer plus loin un paquet (avec un routage incomplet), comme spécifié dans le Source Route, le routeur fixe la valeur du champ Next Hop Pointer à la valeur du champ Source Route Length. Ainsi la destination suivante du paquet sera uniquement basée sur l'adresse de destination (destination address). De même si le bit F est à 0, alors dans les mêmes conditions, le routeur détruira le paquet.

Reserved (6 bits): initialisé à 0 à l'émission, ignoré à la réception.

- Source Route Length (8 bits): c'est le nombre d'éléments/noeuds dans une en-tête de routage SDRP. La longueur de cet en-tête peut être calculée à partir de cette valeur (longueur=SrcRouteLen*16+8). Ce champ ne doit pas excéder la valeur de 24.
Next Hop Pointer (NextHopPtr - 8 bits): il pointe les éléments/noeuds à atteindre. Il est initialisé à 0 pour pointer le premier élément/noeud de Source Route. Quand il est égal au Source Route Length, alors le Source Route est terminé.

Strict/Loose Bit Mask (24 bits): ce masque est utilisé pour prendre une décision d'aiguillage à un noeud. Si la valeur de Next Hop Pointer est N, alors que le $N^{ème}$ bit du Strict/Loose Bit Mask est à 1, cela indique que le prochain noeud est un noeud Strict Source Route Hop. Tandis que s'il est à 0, le prochain noeud est un Loose Route Hop.

- Source Route (multiple de 128 bits): c'est une liste d'adresses IPv6, indiquant le chemin à suivre par le paquet. Le Source Route peut contenir un ensemble d'adresses de types unicast et cluster.

VI.E)En-tête de fragmentation

L'en-tête de fragmentation est utilisé par la source pour envoyer des paquets plus grands que ne peut acheminer le réseau à leurs destinataires. A la différence de la version 4 d' IP, la fragmentation est exécutée seulement par les noeuds source et non plus par les routeurs qui acheminent les paquets le long du chemin. L'en-tête de fragmentation est repéré par une valeur de Next Header égale à 44 juste après le précédent en-tête. Le format est le suivant :

Next Headr (8 bits): identifie le type d'en-tête suivant immédiatement l'en-tête de fragmentation. Les valeurs sont identiques au champ de protocole de IPv4.

Reserved (8 bits), Res (2 bits) : initialisés à 0 à l'émission, ignorés à la réception.

Fragmentation Offset (13 bits): il indique la position du premier octet dans le datagramme total (non fragmenté). Le premier fragment à la place 0. La valeur du champ est un multiple de 8 octets.

M flag (1 bit): si le bit est à 1, il reste un ou des fragments. Tandis que s'il est à 0 il n'y en a plus.

- Identification (32 bits): une valeur assignée au paquet d'origine qui est différente de tous les autres paquets fragmentés récemment avec la même adresse source, les mêmes adresses de destination et valeur du Fragment Next Header. Ce champ permet d'identifier le datagramme pour sécuriser le réassemblage des paquets. Le numéro d'identification est porté par l'en-tête de tous les différents fragments

VI.F)En-tête d'authentification

L'en-tête d'authentification est utilisé pour authentifier et assurer l'intégrité des paquets. La non-répudiation est obtenue par un algorithme d'authentification exécuté sur l'en-tête d'authentification. Mais elle n'est pas obtenue par tous les algorithmes d'authentification exécutés sur cet en-tête. L'en-tête d'authentification est déterminé par la valeur 51du champ Next Header, et a le format suivant :

Next Header (8 bits): identifie le type d'en-tête suivant immédiatement l'en-tête d'authentification. Les valeurs sont identiques à celles du champ de protocole de IPv4.

Authentication Data Length (Auth Data Len - 8 bits): c'est la longueur du champ Authentication Data, multiple de 8 octets.

Reserved (16 bits): initialisé à 0 à l'émission, ignoré à la réception.

Security Association ID (SAID - 32 bits): lorsqu'il est combiné avec l'adresse source, il identifie au(x) destinataire(s) le type de sécurité établi, associé au paquets concernés.

- Authentication Data (longueur variable): information sur l'algorithme spécifique nécessaire à authentifier la source du paquet et à assurer son intégrité conformément à la sécurité associée. La longueur de ce champ est variable et est un multiple de 8 octets.

VI.G)<u>En-tête de confidentialité</u>

L'en-tête privé cherche à donner une confidentialité et une intégrité en cryptant les données à protéger et en les plaçant dans la section données de l'en-tête de confidentialité (Privacy Header). Suivant les exigences de sécurité de l'utilisateur, soit la trame de couche transport (e.g. UDP ou TCP) est cryptée, soit le datagramme entierd'IPv6 l'est. Cette approche par encapsulation est nécessaire pour assurer une confidentialité du datagramme complet original. S'il est présent, l'en-tête de confidentialité est toujours le dernier champ non-crypté dans un paquet.
Le Privacy Header travaille entre stations, entre une station et un gateway(passerelle) de sécurité, ou entre des gateways de sécurité. Ceci permet sans d'importants coûts financiers et de performance d'assurer un réseau digne de confiance en transitant dut rafic sécurisé sur des segments du réseau qui ne le sont pas.

Security Association Identifier (SAID - 32 bits): identifie le type de sécurité au datagramme. Si aucune association de sécurité n'a été établie, la valeur de ce champ est 0x0000. Une association de sécurité est unilatérale. Une communication sécurisée entre deux stations doit avoir normalement deux SAID (un pour chaque sens des échanges). La station destinataire utilise la combinaison de la valeur du SAID et de l'adresse origine pour distinguer la correcte association.

Initialization Vector (longueur dépendant du SAID): ce champ est optionnel et sa valeur dépend du SAID utilisé. Par exemple, le champ peut contenir des données de synchronisation de cryptographie pour un algorithme de codage. Il peut aussi contenir un vecteur d' initialisation cryptographique. L'implantation d'une en-tête de confidentialité utilisera une valeur de SAID pour déterminer si le champ n'est pas vide, et si c'est le cas, évalue la longueur du champ et l'utilise.

Next Header (8 bits), crypté: identifie le type d'en-tête suivant immédiatement l'en-tête de confidentialité. Les valeurs sont identiques à celles du champ de protocole de IPv4.

Reserved (17 bits), crypté: ignoré à la réception.

Length (8 bits), crypté: longueur de l'en-tête privé, à l'exclusion des 8 premiers octets, donnée en un multiple de 8 octets.

Protected Data (longueur variable), crypté: ce champ peut contenir un datagramme complet encapsulé IPv6, une séquence d'option(s) IPv6 ou pas, et enfin le paquet de la couche transport. Ou bien, il est constitué du paquet de la couche transport précédé ou pas d'une série d'option(s).

Algorithme-dependant Trailer (trailer - longueur variable suivant SAID), crypté: ce champ est utilisé pour faire du bourrage (nécessité de certains algorithmes) ou pour enregistrer des données d'authentification à

utiliser avec un algorithme de cryptographie qui fournit la confidentialité sans l'authentification. Ce champ n'est présent que si l'algorithme utilisé le nécessite.

En-tête de bout-en-bout

L'en-tête d'options bout-en-bout donne une information optionnelle qui doit être contrôlée par le(s) noeud(s) destinataire(s) du paquet. L'en-tête des options bout-en-bout est identifié par une valeur de Next Header de TBD suivant immédiatement l'en-tête précédent, et a le même format que l'en-tête d'option noeud-par-noeud, à l'exception de la capacité d'exclure une option de calcul d'intégrité.

Conclusion

A moins d'une entente cordiale entre les organismes de normalisation et les constructeurs, on verra apparaître de nouveaux protocoles imposés par les constructeurs. Les avancées technologiques en matière de transmissions laissent supposer de futurs protocoles pouvant jouir des nouvelles possibilités de traitement. La démocratisation du monde des réseaux rend communs des mots comme IP ou internet. Mais avec cette démocratisation, il faut gérer une demande forte et une qualité de service. Les projets sont donc ambitieux et l'ATM figure comme l'un des précurseurs ; il est adaptable et permet plusieurs débits. C'est dans la pluralité des services que l'ATM se montre aujourd'hui comme le premier des protocoles du futur.

ANNEXES

PROTOCOL NUMBERS

In the Internet Protocol (IP) [45,105] there is a field, called Protocol, to identify the the next level protocol. This is an 8 bit field.

Assigned Internet Protocol Numbers

Decimal	Keyword	Protocol	References
0		Reserved	[JBP]
1	ICMP	Internet Control Message	[97,JBP]
2	IGMP	Internet Group Management	[43,JBP]
3	GGP	Gateway-to-Gateway	[60,MB]
4		Unassigned	[JBP]
5	ST	Stream	[49,JWF]
6	TCP	Transmission Control	[106,JBP]
7	UCL	UCL	[PK]
8	EGP	Exterior Gateway Protocol	[123,DLM1]
9	IGP	any private interior gateway	[JBP]
10	BBN-RCC-MON	BBN RCC Monitoring	[SGC]
11	NVP-II	Network Voice Protocol	[22,SC3]
12	PUP	PUP	[8,XEROX]
13	ARGUS	ARGUS	[RWS4]
14	EMCON	EMCON	[BN7]
15	XNET	Cross Net Debugger	[56,JFH2]
16	CHAOS	Chaos	[NC3]
17	UDP	User Datagram	[104,JBP]
18	MUX	Multiplexing	[23,JBP]
19	DCN-MEAS	DCN Measurement Subsystems	[DLM1]
20	HMP	Host Monitoring	[59,RH6]
21	PRM	Packet Radio Measurement	[ZSU]
22	XNS-IDP	XEROX NS IDP	[133,XEROX]
23	TRUNK-1	Trunk-1	[BWB6]
24	TRUNK-2	Trunk-2	[BWB6]
25	LEAF-1	Leaf-1	[BWB6]
26	LEAF-2	Leaf-2	[BWB6]
27	RDP	Reliable Data Protocol	[138,RH6]
28	IRTP	Internet Reliable Transaction	[79,TXM]
29	ISO-TP4	ISO Transport Protocol Class 4	[63,RC77]

30	NETBLT	Bulk Data Transfer Protocol	[20,DDC1]
31	MFE-NSP	MFE Network Services Protocol	[124,BCH2]
32	MERIT-INP	MERIT Internodal Protocol	[HWB]
33	SEP	Sequential Exchange Protocol	[JC120]
34	3PC	Third Party Connect Protocol	[SAF3]
35-60		Unassigned	[JBP]
61		any host internal protocol	[JBP]
62	CFTP	CFTP	[50,HCF2]
63		any local network	[JBP]
64	SAT-EXPAK	SATNET and Backroom EXPAK	[SHB]
65		Unassigned	[JBP]
66	RVD	MIT Remote Virtual Disk Protocol	[MBG]
67	IPPC	Internet Pluribus Packet Core	[SHB]
68		any distributed file system	[JBP]
69	SAT-MON	SATNET Monitoring	[SHB]
70	VISA	VISA Protocol	[GXT1]
71	IPCV	Internet Packet Core Utility	[SHB]
72-75		Unassigned	[JBP]
76	BR-SAT-MON	Backroom SATNET Monitoring	[SHB]
77	SUN-ND	SUN ND PROTOCOL-Temporary	[WM3]
78	WB-MON	WIDEBAND Monitoring	[SHB]
79	WB-EXPAK	WIDEBAND EXPAK	[SHB]
80	ISO-IP	ISO Internet Protocol	[MTR]
81	VMTP	VMTP	[DRC3]
82	SECURE-VMTP	SECURE-VMTP	[DRC3]
83	VINES	VINES	[BXH]
84	TTP	TTP	[JXS]
85	NSFNET-IGP	NSFNET-IGP	[HWB]
86	DGP	Dissimilar Gateway Protocol	[74,ML109]
87	TCF	TCF	[GAL5]
88	IGRP	IGRP	[18,GXS]
89	OSPFIGP	OSPFIGP	[83,JTM4]
90	Sprite-RPC	Sprite RPC Protocol	[143,BXW]
91	LARP	Locus Address Resolution Protocol	[BXH]
92-254		Unassigned	[JBP]
255		Reserved	[JBP]

PORT NUMBERS

Ports are used in the TCP [45,106] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port".

To the extent possible, these same port assignments are used with the UDP [46,104].

To the extent possible, these same port assignments are used with the ISO-TP4 [64].

The assigned ports use a small portion of the possible port numbers. The assigned ports have all except the low order eight bits cleared to zero. The low order eight bits are specified here.

Port Assignments:

Decimal	Keyword	Description	References
0		Reserved	[JBP]
1	TCPMUX	TCP Port Service Multiplexer	[MKL]
2-4		Unassigned	[JBP]
5	RJE	Remote Job Entry	[12,JBP]
7	ECHO	Echo	[95,JBP]
9	DISCARD	Discard	[94,JBP]
11	USERS	Active Users	[89,JBP]
13	DAYTIME	Daytime	[93,JBP]
15		Unassigned	[JBP]
17	QUOTE	Quote of the Day	[100,JBP]
19	CHARGEN	Character Generator	[92,JBP]
20	FTP-DATA	File Transfer [Default Data]	[96,JBP]
21	FTP	File Transfer [Control]	[96,JBP]
23	TELNET	Telnet	[112,JBP]
25	SMTP	Simple Mail Transfer	[102,JBP]
27	NSW-FE	NSW User System FE	[24,RHT]
29	MSG-ICP	MSG ICP	[85,RHT]
31	MSG-AUTH	MSG Authentication	[85,RHT]
33	DSP	Display Support Protocol	[EXC]
35		any private printer server	[JBP]
37	TIME	Time	[108,JBP]
39	RLP	Resource Location Protocol	[MA]
41	GRAPHICS	Graphics	[129,JBP]
42	NAMESERVER	Host Name Server	[99,JBP]
43	NICNAME	Who Is	[55,MARY]
44	MPM-FLAGS	MPM FLAGS Protocol	[JBP]
45	MPM	Message Processing Module [recv]	[98,JBP]
46	MPM-SND	MPM [default send]	[98,JBP]
47	NI-FTP	NI FTP	[134,SK8]
49	LOGIN	Login Host Protocol	[PHD1]
51	LA-MAINT	IMP Logical Address Maintenance	[76,AGM]
53	DOMAIN	Domain Name Server	[81,95,PM1]
55	ISI-GL	ISI Graphics Language	[7,RB9]
57		any private terminal access	[JBP]

59		any private file service	[JBP]
61	NI-MAIL	NI MAIL	[5,SK8]
63	VIA-FTP	VIA Systems - FTP	[DXD]
65	TACACS-DS	TACACS-Database Service	[3,KH43]
67	BOOTPS	Bootstrap Protocol Server	[36,WJC2]
68	BOOTPC	Bootstrap Protocol Client	[36,WJC2]
69	TFTP	Trivial File Transfer	[126,DDC1]
71	NETRJS-1	Remote Job Service	[10,RTB3]
72	NETRJS-2	Remote Job Service	[10,RTB3]
73	NETRJS-3	Remote Job Service	[10,RTB3]
74	NETRJS-4	Remote Job Service	[10,RTB3]
75		any private dial out service	[JBP]
77		any private RJE service	[JBP]
79	FINGER	Finger	[52,KLH]
81	HOSTS2-NS	HOSTS2 Name Server	[EAK1]
83	MIT-ML-DEV	MIT ML Device	[DPR]
85	MIT-ML-DEV	MIT ML Device	[DPR]
87		any private terminal link	[JBP]
89	SU-MIT-TG	SU/MIT Telnet Gateway	[MRC]
91	MIT-DOV	MIT Dover Spooler	[EBM]
93	DCP	Device Control Protocol	[DT15]
95	SUPDUP	SUPDUP	[27,MRC]
97	SWIFT-RVF	Swift Remote Vitural File Protocol	[MXR]
98	TACNEWS	TAC News	[ANM2]
99	METAGRAM	Metagram Relay	[GEOF]
101	HOSTNAME	NIC Host Name Server	[54,MARY]
102	ISO-TSAP	ISO-TSAP	[16,MTR]
103	X400	X400	[HCF2]
104	X400-SND	X400-SND	[HCF2]
105	CSNET-NS	Mailbox Name Nameserver	[127,MS56]
107	RTELNET	Remote Telnet Service	[101,JBP]
109	POP2	Post Office Protocol - Version 2	[14,JKR1]
110	POP3	Post Office Protocol - Version 3	[122,MTR]
111	SUNRPC	SUN Remote Procedure Call	[DXG]
113	AUTH	Authentication Service	[130,MCSJ]
115	SFTP	Simple File Transfer Protocol	[73,MKL1]
117	UUCP-PATH	UUCP Path Service	[44,MAE]
119	NNTP	Network News Transfer Protocol	[65,PL4]
121	ERPC	Encore Expedited Remote Proc. Call	[132,JXO]
123	NTP	Network Time Protocol	[80,DLM1]
125	LOCUS-MAP	Locus PC-Interface Net Map Server	[137,EP53]
127	LOCUS-CON	Locus PC-Interface Conn Server	[137,EP53]
129	PWDGEN	Password Generator Protocol	[141,FJW]
130	CISCO-FNA	CISCO FNATIVE	[WXB]
131	CISCO-TNA	CISCO TNATIVE	[WXB]
132	CISCO-SYS	CISCO SYSMAINT	[WXB]
133	STATSRV	Statistics Service	[DLM1]
134	INGRES-NET	INGRES-NET Service	[MXB]
135	LOC-SRV	Location Service	[JXP]
136	PROFILE	PROFILE Naming System	[LLP]
137	NETBIOS-NS	NETBIOS Name Service	[JBP]
138	NETBIOS-DGM	NETBIOS Datagram Service	[JBP]
139	NETBIOS-SSN	NETBIOS Session Service	[JBP]
140	EMFIS-DATA	EMFIS Data Service	[GB7]
141	EMFIS-CNTL	EMFIS Control Service	[GB7]
142	BL-IDM	Britton-Lee IDM	[SXS1]
143	IMAP2	Interim Mail Access Protocol v2	[MRC]
144	NEWS	NewS	[JAG]

145	UAAC	UAAC Protocol	[DAG4]
146	ISO-TP0	ISO-IP0	[86,MTR]
147	ISO-IP	ISO-IP	[MTR]
148	CRONUS	CRONUS-SUPPORT	[135,JXB]
149	AED-512	AED 512 Emulation Service	[AXB]
150	SQL-NET	SQL-NET	[MXP]
151	HEMS	HEMS	[87,CXT]
152	BFTP	Background File Transfer Program	[AD14]
153	SGMP	SGMP	[37,MS9]
154	NETSC-PROD	NETSC	[SH37]
155	NETSC-DEV	NETSC	[SH37]
156	SQLSRV	SQL Service	[CMR]
157	KNET-CMP	KNET/VM Command/Message Protocol	[77,GSM11]
158	PCMail-SRV	PCMail Server	[19,MXL]
159	NSS-Routing	NSS-Routing	[JXR]
160	SGMP-TRAPS	SGMP-TRAPS	[37,MS9]
161	SNMP	SNMP	[15,MTR]
162	SNMPTRAP	SNMPTRAP	[15,MTR]
163	CMIP-Manage	CMIP/TCP Manager	[4,AXB1]
164	CMIP-Agent	CMIP/TCP Agent	[4,AXB1]
165	XNS-Courier	Xerox	[144,SXA]
166	S-Net	Sirius Systems	[BXL]
167	NAMP	NAMP	[MS9]
168	RSVD	RSVD	[NT12]
169	SEND	SEND	[WDW11]
170	Print-SRV	Network PostScript	[BKR]
171	Multiplex	Network Innovations Multiplex	[KXD]
172	CL/1	Network Innovations CL/1	[KXD]
173	Xyplex-MUX	Xyplex	[BXS]
174	MAILQ	MAILQ	[RXZ]
175	VMNET	VMNET	[CXT]
176	GENRAD-MUX	GENRAD-MUX	[RXT]
177	XDMCP	X Display Manager Control Protocol	[RWS4]
178	NextStep	NextStep Window Server	[LXH]
179	BGP	Border Gateway Protocol	[KSL]
180	RIS	Intergraph	[DXB]
181	Unify	Unify	[VXS]
182	Unisys-Cam	Unisys-Cam	[GXG]
183	OCBinder	OCBinder	[JXO1]
184	OCServer	OCServer	[JXO1]
185	Remote-KIS	Remote-KIS	[RXD1]
186	KIS	KIS Protocol	[RXD1]
187	ACI	Application Communication Interface	[RXC1]
188	MUMPS	MUMPS	[HS23]
189	QFT	Queued File Transport	[WXS]
190	GACP	Gateway Access Control Protocol	[PCW]
191	Prospero	Prospero	[BCN]
192	OSU-NMS	OSU Network Monitoring System	[DXK]
193	SRMP	Spider Remote Monitoring Protocol	[TXS]
194	IRC	Internet Relay Chat Protocol	[JXO2]
195	DN6-NLM-AUD	DNSIX Network Level Module Audit	[LL69]
196	DN6-SMM-RED	DNSIX Session Mgt Module Audit Redirect	[LL69]
197	DLS	Directory Location Service	[SXB]
198	DLS-Mon	Directory Location Service Monitor	[SXB]
198-200		Unassigned	[JBP]
201	AT-RMTP	AppleTalk Routing Maintenance	[RXC]
202	AT-NBP	AppleTalk Name Binding	[RXC]
203	AT-3	AppleTalk Unused	[RXC]

204	AT-ECHO	AppleTalk Echo	[RXC]
205	AT-5	AppleTalk Unused	[RXC]
206	AT-ZIS	AppleTalk Zone Information	[RXC]
207	AT-7	AppleTalk Unused	[RXC]
208	AT-8	AppleTalk Unused	[RXC]
209-223		Unassigned	[JBP]
224-241		Reserved	[JBP]
243	SUR-MEAS	Survey Measurement	[6,DDC1]
245	LINK	LINK	[1,RDB2]
246	DSP3270	Display Systems Protocol	[39,WJS1]
247-255		Reserved	[JBP]