Encapsulation

Couches 2 à 4

Contenus

Articles

Références

Licence des articles

Ethernet

L'article doit être débarrassé d'une partie de son jargon.

Sa qualité peut être largement améliorée en utilisant un vocabulaire plus directement compréhensible. Discutez des points à améliorer en page de discussion.

Cet article ne cite pas suffisamment ses sources (septembre 2012).

Si vous disposez d'ouvrages ou d'articles de référence ou si vous connaissez des sites web de qualité traitant du thème abordé ici, merci de compléter l'article en donnant les références utiles à sa vérifiabilité et en les liant à la section « Notes et références » (modifier l'article [1]).

Connecteur RJ45 pour Ethernet

Pile de protocoles
7. Application 6. Présentation 5. Session 4. Transport 3. Réseau 2. Liaison 1. Physique
Modèle Internet
Modèle OSI
modifier [2]

Ethernet est un protocole de réseau local à commutation de paquets. Bien qu'il implémente la couche physique (PHY) et la sous-couche *Media Access Control* (MAC) du modèle IEEE 802.3, le protocole Ethernet est classé dans les couche de liaison de données (niveau 2) et physique (niveau 1), puisque la couche LLC (Logical Link Control) 802.2 fait la charnière entre les couches supérieures et la sous-couche MAC (Media Access Control) qui fait partie intégrante du processus 802.3 avec la couche physique, car les formats de trames que le standard définit sont normalisés et peuvent être encapsulés dans des protocoles autres que ses propres couches physiques MAC et PHY. Ces couches physiques font l'objet de normes séparées en fonction des débits, du support de transmission, de la longueur des liaisons et des conditions environnementales.

Ethernet a été standardisé sous le nom IEEE 802.3[3] :

- Ethernet : les 13e et 14e octets d'une trame Ethernet contiennent le type (numéro) de protocole de la couche supérieure (ARP, IPv4, IPv6...) ; comme il n'y a pas d'indication sur la longueur des données, il n'y a pas de couche LLC (Logical Link Control) pour supprimer un bourrage potentiel ⇒ ce sera donc à la couche supérieure (Réseau) de supprimer le bourrage s'il y en a.
- 802.3 : les 13e et 14e octets d'une trame 802.3 contiennent la longueur de la partie des données qui sera gérée par la couche LLC qui, située entre la couche MAC et la couche Réseau, supprimera le bourrage (congestion)[Quoi ?] avant de l'envoyer à la couche Réseau.

C'est maintenant une norme internationale : *ISO/IEC 8802-3*.

Depuis les années 1990, on utilise très fréquemment Ethernet sur paires torsadées pour la connexion des postes clients, et des versions sur fibre optique pour le cœur du réseau. Cette configuration a largement supplanté d'autres standards comme le *Token Ring*, FDDI et ARCNET. Depuis quelques années, les variantes sans-fil d'Ethernet (normes IEEE 802.11, dites « Wi-Fi ») ont connu un fort succès, aussi bien sur les installations personnelles que professionnelles.

Dans un réseau Ethernet, le câble diffuse les données à toutes les machines connectées, de la même façon que les ondes radiofréquences parviennent à tous les récepteurs. Le nom *Ethernet* dérive de cette analogie[4] : avant le XXe siècle on imaginait que les ondes se propageaient dans l'*éther*, milieu hypothétique censé baigner l'Univers. Quant au suffixe *net*, il s'agit de l'abréviation du mot *network* (réseau) en anglais.

Histoire

Ethernet a originellement été développé comme l'un des projets pionniers du Xerox PARC. Une histoire commune veut qu'il ait été inventé en 1973, date à laquelle Robert Metcalfe écrivit un mémo à ses patrons à propos du potentiel d'Ethernet. Metcalfe affirme qu'Ethernet a en fait été inventé sur une période de plusieurs années.[réf. souhaitée] En 1976, Robert Metcalfe et David Boggs (l'assistant de Metcalfe) ont publié un document intitulé *Ethernet: Distributed Packet-Switching For Local Computer Networks* (Ethernet : commutation de paquets distribuée pour les réseaux informatiques locaux).

Metcalfe a quitté Xerox en 1979 pour promouvoir l'utilisation des ordinateurs personnels et des réseaux locaux, et a fondé l'entreprise 3Com. Il réussit à convaincre DEC, Intel et Xerox de travailler ensemble pour promouvoir Ethernet en tant que standard, au terme d'une période au cours de laquelle la réflexion des constructeurs s'oriente vers une informatique décentralisée.

Ethernet était à l'époque en compétition avec deux systèmes propriétaires, *Token Ring* (IBM, plus récent) et ARCnet (TRW-Matra, plus ancien); ces deux systèmes ont au fil du temps diminué en popularité face à l'Ethernet, en raison de la baisse de coûts due à la production de masse. Ethernet avait par ailleurs moins de contraintes topologiques que le token-ring (au CeBIT de 1995, on pouvait voir à titre expérimental un simili plafond blanc utilisé comme medium Internet, les signaux transitant par infrarouge). Pendant ce temps, 3Com est devenue une compagnie majeure du domaine des réseaux informatiques.

Description générale

L'Ethernet est basé sur le principe de membres (pairs) sur le réseau, envoyant des messages dans ce qui était essentiellement un système radio, captif à l'intérieur d'un fil ou d'un canal commun, parfois appelé *l'éther*. Chaque pair est identifié par une clé globalement unique, appelée adresse MAC, pour s'assurer que tous les postes sur un réseau Ethernet aient des adresses distinctes.

Une technologie connue sous le nom de *Carrier Sense Multiple Access with Collision Detection* (Écoute de porteuse avec accès multiples et détection de collision) ou CSMA/CD régit la façon dont les postes accèdent au média. Au départ développée durant les années 1960 pour ALOHAnet à Hawaï en utilisant la radio, la technologie est relativement simple comparée à *Token Ring* ou aux réseaux contrôlés par un maître. Lorsqu'un ordinateur veut envoyer de l'information, il obéit à l'algorithme suivant :

Procédure principale

1. Trame prête à être transmise.
2. Si le medium n'est pas libre, attendre jusqu'à ce qu'il le devienne puis attendre la durée intertrame (9,6 µs pour l'Ethernet 10 Mbps) et démarrer la transmission.
3. Si une collision est détectée :
   - lancer la procédure de gestion des collisions.
   - sinon, la transmission est réussie.

Procédure de gestion des collisions

1. Continuer la transmission à hauteur de la durée d'une trame de taille minimale (64 octets) pour s'assurer que toutes les stations détectent la collision.
2. Si le nombre maximal de transmissions (16) est atteint, annuler la transmission.
3. Attendre un temps aléatoire dépendant du nombre de tentatives de transmission.
4. Reprendre la procédure principale.

En pratique, ceci fonctionne comme une discussion ordinaire, où les gens utilisent tous un médium commun (l'air) pour parler à quelqu'un d'autre. Avant de parler, chaque personne attend poliment que plus personne ne parle. Si deux personnes commencent à parler en même temps, les deux s'arrêtent et attendent un court temps aléatoire. Il y a de bonnes chances que les deux personnes attendent un délai différent, évitant donc une autre collision. Des temps d'attente en progression exponentielle sont utilisés lorsque plusieurs collisions surviennent à la suite.

Comme dans le cas d'un réseau non commuté, toutes les communications sont émises sur un médium partagé, toute information envoyée par un poste est reçue par tous les autres, même si cette information était destinée à une seule personne. Les ordinateurs connectés sur l'Ethernet doivent donc filtrer ce qui leur est destiné ou non. Ce type de communication « quelqu'un parle, tous les autres entendent » d'Ethernet est une de ses faiblesses, car, pendant que l'un des nœuds émet, toutes les machines du réseau reçoivent et doivent, de leur côté, observer le silence. Ce qui fait qu'une communication à fort débit entre seulement deux postes peut saturer tout un réseau local.

De même, comme les chances de collision sont proportionnelles au nombre de transmetteurs et aux données envoyées, le réseau devient extrêmement congestionné au-delà de 50 % de sa capacité (indépendamment du nombre de sources de trafic). Pour résoudre ce problème, les commutateurs ont été développés afin de maximiser la bande passante disponible.

Suivant le débit utilisé, il faut tenir compte du domaine de collision régi par les lois de la physique et notamment le déplacement électronique dans un câble de cuivre. Si l'on ne respecte pas ces distances maximales entre machines, le protocole CSMA/CD n'a pas lieu d'exister.

De même si on utilise un commutateur, CSMA/CD est désactivé. Et ceci pour une raison que l'on comprend bien. Avec CSMA/CD, on écoute ce que l'on émet, si quelqu'un parle en même temps que moi il y a collision. Il y a donc incompatibilité avec le mode *full-duplex* des commutateurs.

Types de trames Ethernet et champ *EtherType*

Article détaillé : EtherType.

Il y a quatre types de trame Ethernet :

- Ethernet originale version I (n'est plus utilisée)
- Ethernet Version 2 ou Ethernet II (appelée trame DIX, toujours utilisée)
- IEEE 802.x LLC
- IEEE 802.x LLC/SNAP

Ces différents types de trame ont des formats et des valeurs de *MTU* différents mais peuvent coexister sur un même médium physique.

La version 1 originale de Xerox possède un champ de 16 bits identifiant la taille de trame, même si la longueur maximale d'une trame était de 1 500 octets. Ce champ fut vite réutilisé dans la version 2 de Xerox comme champ d'identification, avec la convention que les valeurs entre 0 et 1 500 indiquaient une trame Ethernet originale, mais que les valeurs plus grandes indiquaient ce qui a été appelé l'EtherType, et l'utilisation du nouveau format de trame. Cette utilisation duale du même champ de données justifie son appellation courante de champ longueur/type. En résumé, si x est la valeur dudit champ :

- x ≤ 1 500 : trame Ethernet I
- x ≥ 1 501 : trame Ethernet II

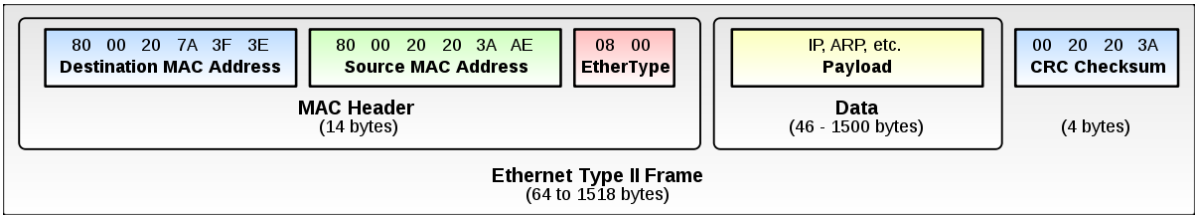
L'IEEE 802.3 a de nouveau défini le champ de 16 bits après les adresses MAC comme la longueur. Comme l'Ethernet I n'est plus utilisé, ceci permet désormais aux logiciels de déterminer si une trame est de type Ethernet II ou IEEE 802.3, permettant la cohabitation des deux standards sur le même médium physique. Toutes les trames 802.3 ont un champ LLC. En examinant ce dernier, il est possible de déterminer s'il est suivi par un champ SNAP ou non. La convention en vigueur actuellement est donc, si x est la valeur du champ longueur/type :

- x ≤ 1 500 : trame 802.3 avec LLC (et éventuellement SNAP)
- x ≥ 1 536 : trame Ethernet II

Les valeurs entre 1 500 et 1 536 sont indéfinies et ne devraient jamais être employées.

Synthèse graphique

Les différentes trames peuvent coexister sur un même réseau physique.



La trame Ethernet de format : type II

Information extraite du document de G.Requilé du CNRS et adaptée

Trame Ethernet II

En octets

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14 … 1513	1514	1515	1516	1517
Adresse MAC destination						Adresse MAC source						Type de protocole		Données	FCS/CRC			

Attention il existe d'autres types de trames Ethernet qui possèdent d'autres particularités. Le champ *Type de protocole* peut prendre par exemple les valeurs suivantes :

Article détaillé : Service Access Point.

- 0x0800 : IPv4
- 0x86DD : IPv6
- 0x0806 : ARP
- 0x8035 : RARP
- 0x809B : AppleTalk
- 0x88CD : SERCOS III
- 0x0600 : XNS
- 0x8100 : VLAN

Remarques :

- comme expliqué ci-dessus, si le champ **type de protocole** possède une valeur hexadécimale inférieure à 0x05DC alors la trame est une trame Ethernet 802.3 et ce champ indique la longueur du champ *données* ;
- on notera la présence parfois d'un préambule de 64 bits de synchronisation, alternance de 1 et 0 avec les deux derniers bits à 1 (non représenté sur la trame) ;
- l'adresse de *broadcast* (*diffusion*) Ethernet a tous ses bits à *1* ;
- la taille minimale des données est de 46 octets (RFC 894 - Frame Format).
- si nécessaire, pour atteindre les 46 octets de données, un bourrage est effectué, et celui-ci est transparent au niveau utilisateur[5].

Variétés d'Ethernet

La section ci-dessous donne un bref résumé de tous les types de média d'Ethernet. En plus de tous ces standards officiels, plusieurs fabricants ont implémenté des types de média propriétaires pour différentes raisons—quelquefois pour supporter de plus longues distances sur de la fibre optique.

Quelques anciennes variétés d'Ethernet

- Xerox Ethernet -- L'implémentation originale d'Ethernet, qui a eu deux versions, la version 1 et 2, durant son développement. La version 2 est encore souvent utilisée.
- 10BASE5 (aussi appelé *Thick Ethernet*) -- Ce standard de l'IEEE publié très tôt utilise un câble coaxial simple dans lequel on insère une connexion en perçant le câble pour se connecter au centre et à la masse (prises *vampires*). Largement désuet, mais à cause de plusieurs grandes installations réalisées très tôt, quelques systèmes peuvent encore être en utilisation.
- 10BROAD36 -- Obsolète. Un vieux standard supportant l'Ethernet sur de longues distances. Il utilisait des techniques de modulation en large bande similaires à celles employées par les modems câble, opérées sur un câble coaxial.
- 1BASE5 -- Une tentative de standardisation de solution pour réseaux locaux à bas prix. Il opère à 1 Mbit/s mais a été un échec commercial.

Ethernet 10 Mbit/s

- 10BASE2 (aussi appelé *ThinNet* ou *Cheapernet*) -- un câble coaxial de 50 Ohms connecte les machines ensemble, chaque machine utilisant un adaptateur en T pour se brancher à sa carte réseau. Requiert une terminaison à chaque bout. Pendant plusieurs années, ce fut le standard Ethernet dominant.
- 10BASE-T -- Fonctionne avec minimum 4 fils (deux paires torsadées, conventionnellement les 1, 2 et 3, 6) sur un câble CAT-3 ou CAT-5 avec connecteur RJ45. Un concentrateur (ou *hub*) ou un commutateur (ou *switch*) est au centre du réseau, ayant un port pour chaque nœud. C'est aussi la configuration utilisée pour le 100BASE-T et le Gigabit Ethernet (câble CAT-6). Bien que la présence d'un nœud central (le *hub*) donne une impression visuelle de topologie en étoile, il s'agit pourtant bien d'une topologie en bus - tous les signaux émis sont reçus par l'ensemble des machines connectées. La topologie en étoile n'apparaît que si on utilise un commutateur (*switch*).
- FOIRL -- *Fiber-optic inter-repeater link* (lien inter-répéteur sur fibre optique). Le standard original pour l'Ethernet sur la fibre optique.
- 10BASE-F -- Terme générique pour la nouvelle famille d'Ethernet 10 Mbit/s : 10BASE-FL, 10BASE-FB et 10BASE-FP. De ceux-ci, seulement 10BASE-FL est beaucoup utilisé.
- 10BASE-FL -- Une mise à jour du standard FOIRL.
- 10BASE-FB -- Prévu pour inter-connecter des concentrateurs ou commutateurs au cœur du réseau, mais maintenant obsolète.
- 10BASE-FP -- Un réseau en étoile qui ne nécessitait aucun répéteur, mais qui n'a jamais été réalisé.

Fast Ethernet (100 Mbit/s)

- 100BASE-T -- Un terme pour n'importe lequel des standards 100 Mbit/s sur paire torsadée. Inclut 100BASE-TX, 100BASE-T4 et 100BASE-T2.
- 100BASE-TX -- Utilise deux paires et requiert du câble CAT-5. Topologie en bus en utilisant un concentrateur (*hub*) ou en étoile avec un commutateur (*switch*), comme pour le 10BASE-T, avec lequel il est compatible.
- 100BASE-T4 -- Permet le 100 Mbit/s (en semi-duplex seulement) sur du câble CAT-3 (qui était utilisé dans les installations 10BASE-T). Utilise les quatre paires du câble. Maintenant désuet, comme le CAT-5 est la norme actuelle.
- 100BASE-T2 -- Aucun produit n'existe. Supporte le mode *full-duplex* et utilise seulement deux paires, avec des câbles CAT-3. Il est équivalent au 100BASE-TX sur le plan des fonctionnalités, mais supporte les vieux câbles.
- 100BASE-FX -- Ethernet 100 Mbit/s sur fibre optique.

Gigabit Ethernet (1 000 Mbit/s)

- 1000BASE-T -- 1 Gbit/s sur câble de paires torsadées de catégorie 5 (classe D) ou supérieure (selon NF EN 50173-2002), sur une longueur maximale de 100 m. Utilise les 4 paires en *full duplex*, chaque paire transmettant 2 bits par *top* d'horloge, à l'aide d'un code à 5 moments. Soit un total de 1 octet sur l'ensemble des 4 paires, dans chaque sens. Compatible avec 100BASE-TX et 10BASE-T, avec détection automatique des Tx et Rx assurée. La topologie est ici toujours en étoile car il n'existe pas de concentrateurs 1 000 Mbit/s. On utilise donc obligatoirement des commutateurs (*switch*).
- 1000BASE-X -- 1 Gbit/s qui utilise des interfaces modulaires (appelés GBIC) adaptées au média (Fibre Optique Multi, Mono-mode, cuivre).
- 1000BASE-SX -- 1 Gbit/s sur fibre optique multimodes à 850 nm.
- 1000BASE-LX -- 1 Gbit/s sur fibre optique monomodes et multimodes à 1 300 nm.
- 1000BASE-LH -- 1 Gbit/s sur fibre optique, sur longues distances.
- 1000BASE-ZX -- 1 Gbit/s sur fibre optique monomodes longues distances.
- 1000BASE-CX -- Une solution pour de courtes distances (jusqu'à 25 m) pour le 1 Gbit/s sur du câble de cuivre spécial.

(cf. cercle CREDO)

Ethernet 10 gigabit par seconde

Le nouveau standard Ethernet 10 Gigabits entoure sept types de média différents pour les réseaux locaux, réseaux métropolitains et réseaux étendus. Il est actuellement spécifié par un standard supplémentaire, l'IEEE 802.3ae dont la première publication date de 2002, et va être incorporé dans une révision future de l'IEEE 802.3. La version Ethernet 10 Gbit/s est 10 fois plus rapide que Gigabit Ethernet ; ceci est vrai jusqu'au niveau de la couche MAC seulement.

- 10GBASE-CX4 (cuivre, câble *infiniband*, 802.3ak) -- utilise un câble en cuivre de type *infiniband* 4x sur une longueur maximale de 15 mètres.
- 10GBASE-T -- transmission sur câble catégorie 6, 6 A ou 7 (802.3an), en full duplex sur 4 paires avec un nombre de moments de codage qui sera fonction de la catégorie retenue pour le câble (et de l'immunité au bruit souhaitée), sur une longueur maximale de 100 mètres. Devrait être compatible avec 1000BASE-T, 100BASE-TX et 10BASE-T
- 10GBASE-SR (850 nm MM, 300 mètres, *dark fiber*) -- créé pour supporter de courtes distances sur de la fibre optique multimode, il a une portée de 26 à 82 mètres, en fonction du type de câble. Il supporte aussi les distances jusqu'à 300 m sur la nouvelle fibre multimode 2 000 MHz.
- 10GBASE-LX4 -- utilise le multiplexage par division de longueur d'onde pour supporter des distances entre 240 et 300 mètres sur fibre multimode.
- 10GBASE-LR (1 310 nm SM, 10 km, *dark fiber*) et 10GBASE-ER (1 550 nm SM, 40 km, *dark fiber*) -- Ces standards supportent jusqu'à 10 et 40 km respectivement, sur fibre monomode.
- 10GBASE-SW (850 nm MM, 300 mètres, SONET), 10GBASE-LW (1 310 nm SM, 10 km, SONET) et 10GBASE-EW (1 550 nm SM, 40 km, SONET). Ces variétés utilisent le *WAN PHY*, étant conçu pour inter-opérer avec les équipements OC-192 / STM-64 SONET/SDH. Elles correspondent au niveau physique à 10GBASE-SR, 10GBASE-LR et 10GBASE-ER respectivement, et utilisent le même type de fibre, en plus de supporter les mêmes distances. (Il n'y a aucun standard *WAN PHY* correspondant au 10GBASE-LX4.)

L'Ethernet 10 Gigabits est assez récent, et il reste à voir lequel des standards va obtenir l'acceptation des compagnies.

Mode LAN et mode WAN

10 Gigabit Ethernet supporte seulement le mode *full duplex*, beaucoup de liens sont en mode point à point bien que du routage à ce débit commence à apparaître. Le **mode LAN** fonctionne à un débit ligne, au niveau de la fibre, de 10,3 Gbit/s ce qui représente en fait le débit MAC de 10 Gbit/s pondéré par 66/64 rapport lié au codage de la couche PCS utilisant un code de ligne 64B66B. Le sur-débit de ce code est de 3 %, à comparer aux 25 % du code 8B10B du mode Gigabit Ethernet.

L'importance du **mode WAN** PHY est incontestable et permet de transporter les trames Ethernet 10 Gigabits sur des liens SDH ou SONET actuellement en place dans beaucoup de réseaux. Le mode WAN PHY opère à un débit légèrement inférieur à 10Gbe, à savoir 9 953 280 kbit/s (ce qui correspond au débit STM64/OC192). Le conteneur virtuel 64c ou 192c véhicule des codes 64B66B.

Les modules optiques : couche PMD (PHY).
Divers fabricants (Fiberxon, Sumitomo, Finisar, etc.) proposent des modules XFP, normalisés selon le XFP MSA Group, permettant une interopérabilité. Ces modules permettent de convertir le signal optique (côté ligne) en un signal électrique différentiel (côté matériel) au débit de 10,3 Gbit/s; c'est donc l'équivalent de la couche PHY au niveau PMD du modèle OSI.

Les serdes : couche PMA (PHY).
Ce signal de 10 Gb/s, trop rapide, ne peut pas être traité directement, il faut donc le paralléliser, en général sur 64 bits. Des circuits dédiés spécialisés permettent cette conversion.

Le mot serdes vient de l'anglais pour *serialiser/deserialiser*.

Le codage 64B66B : couche PCS (PHY)

Le code en ligne utilisé 64B66B transforme le format XGMII (64 bits de données plus 8 bits de contrôle) en mots de 66 bits. L'objectif est multiple :

- apporter une dispersion d'énergie et éviter de longues suites consécutives de '0' ou '1' que les XFP peuvent ne pas trop apprécier.
- ceci apporte donc des transitions afin de faciliter les mécanismes de récupération d'horloge.

Le code 66 bits est composé de deux bits de synchronisation suivis de 64 bits de donnée.

- Si la synchro est '01', les 64 bits sont de type donnée
- Si la synchro est '10', les 64 bits contiennent au moins un octet de contrôle
- Les préambules '00' et '11' ne sont pas utilisés.

Les 64 bits de données sont embrouillés par un embrouilleur auto synchronisé.

À ce niveau-là nous retrouvons un format équivalent MII, les couches suivantes : *data link* (MAC), *network* (IP), *transport* (TCP/UDP) fonctionnant de façon similaire à gigabit Ethernet.

Notes et références

[1] http://fr.wikipedia.org/w/index.php?title=Ethernet&action=edit
[2] http://fr.wikipedia.org/w/index.php?title=Ethernet&action=edit&section=0
[3] IEEE 802.3 ETHERNET WORKING GROUP (http://www.ieee802.org/3/), sur le site ieee802.org
[4] (http://books.google.fr/books?id=VbTvnxBKzUgC&dq=Ethernet:+The+Definitive+Guide&pg=PP1&ots=HXGXgOGxaY&sig=C-3WhkefLUk8HAscMmBvFiz37KA&hl=fr&sa=X&oi=book_result&resnum=1&ct=result#PPA65,M1), Charles E. Spurgeon. O'Reilly, 2000. 5, *Invention of Ethernet*.
[5] *Entête Ethernet*, §5.7. (http://www.frameip.com/entete-ethernet/) Frameip.

Address Resolution Protocol

Pour les articles homonymes, voir ARP.

Pile de protocoles
7. Application 6. Présentation 5. Session 4. Transport 3. Réseau 2. Liaison 1. Physique
Modèle Internet
Modèle OSI
modifier [1]

L'***Address resolution protocol*** (ARP, protocole de résolution d'adresse) est un protocole effectuant la traduction d'une adresse de protocole de couche réseau (typiquement une adresse IPv4) en une adresse MAC (typiquement une adresse ethernet), ou même de tout matériel de couche de liaison. Il se situe à l'interface entre la couche réseau (couche 3 du modèle OSI) et la couche de liaison (couche 2 du modèle OSI).

Il a été défini dans la RFC 826 : *An Ethernet Address Resolution Protocol*.

Le protocole ARP est nécessaire au fonctionnement d'IPv4 utilisé au-dessus d'un réseau de type ethernet. En IPv6, les fonctions de ARP sont reprises par le *Neighbor Discovery Protocol* (NDP).

Dans la suite de l'article, le terme *adresse IP* est utilisé pour parler d'adresse IPv4.

Fonctionnement

Un ordinateur connecté à un réseau informatique souhaite émettre une trame ethernet à destination d'un autre ordinateur dont il connaît l'adresse IP et placé dans le même sous-réseau. Dans ce cas, cet ordinateur va placer son émission en attente et effectuer une requête ARP en *broadcast*. Cette requête est de type « quelle est l'adresse MAC correspondant à l'adresse IP *adresseIP* ? Répondez à *adresseIP* ».

Puisqu'il s'agit d'un *broadcast*, tous les ordinateurs du segment vont recevoir la requête. En observant son contenu, ils pourront déterminer quelle est l'adresse IP sur laquelle porte la recherche. La machine qui possède cette adresse IP sera la seule à répondre en envoyant à la machine émettrice une réponse ARP du type « je suis *adresseIP*, mon adresse MAC est *adresseMAC* ». Pour émettre cette réponse au bon ordinateur, il crée une entrée dans son cache ARP à partir des données contenues dans la requête ARP qu'il vient de recevoir.

La machine à l'origine de la requête ARP reçoit la réponse, met à jour son cache ARP et peut donc envoyer le message qu'elle avait mis en attente jusqu'à l'ordinateur concerné.

Il suffit donc d'un *broadcast* et d'un *unicast* pour créer une entrée dans le cache ARP de deux ordinateurs.

Commande arp

La commande `arp` permet la consultation et parfois la modification de la table ARP dans certains systèmes d'exploitation.

- `arp -a` : affiche toutes les entrées dans le cache ARP.
- `arp -a @ip` : dans le cas où il y a plusieurs cartes réseau, on peut faire l'affichage du cache associé à une seule @ip.
- `arp -s @ip @MAC` : ajout manuel d'une entrée statique permanente dans le cache (ce besoin se manifeste si on appelle régulièrement des hôtes, pour réduire le trafic réseau).

Sécurité du protocole ARP

Le protocole ARP a été conçu sans souci particulier de sécurité. Il est vulnérable à des attaques locales sur le segment reposant principalement sur l'envoi de messages ARP erronés à un ou plusieurs ordinateurs. Elles sont regroupées sous l'appellation *ARP poisoning* (pollution de cache ARP). La vulnérabilité d'un ordinateur à la pollution de cache ARP dépend de la mise en œuvre du protocole ARP par son système d'exploitation.

Soit une machine Charlie qui souhaite intercepter les messages d'Alice vers Bob, toutes appartenant au même sous-réseau. L'attaque consiste pour Charlie à envoyer un paquet « `arp who-has` » à la machine d'Alice. Ce paquet spécialement construit contiendra comme IP source, l'adresse IP de la machine de Bob dont nous voulons usurper l'identité (*ARP spoofing*) et l'adresse MAC de la carte réseau de Charlie. La machine d'Alice va ainsi créer une entrée associant notre adresse MAC à l'adresse IP de la machine de Bob. Alice, destinataire de l'« `arp who-has` », utilise le paquet pour créer une entrée dans sa table MAC. Si Alice veut communiquer avec Bob au niveau IP, c'est Charlie qui recevra les trames d'Alice puisque notre adresse MAC est enregistré dans le cache empoisonné de Alice comme équivalence pour l'IP du poste Bob. Ceci est une faiblesse connue de la mise en œuvre d'ARP et permet de corrompre facilement un cache ARP distant.

Ces attaques peuvent permettre une écoute des communications entre deux machines (attaque de l'homme du milieu), le vol de connexion, une surcharge des commutateurs servant de structure au réseau informatique ou un déni de service (il suffit de faire une attaque de type MITM (Man In The Middle) puis de refuser les paquets).

Pour lutter contre ce type d'attaque, il est possible :

- de mettre en place des entrées statiques dans le cache ARP de chaque machine du réseau (commande `arp -s`). Ceci n'est applicable qu'à un faible nombre de machines (on privilégie les plus critiques, comme les serveurs et les passerelles). Sur les systèmes d'exploitation Microsoft Windows antérieurs à la version XP, une entrée statique peut être mise à jour, la seule différence est qu'elle n'expire pas ;
- de limiter les adresses MAC sur chaque port (renseignement statique) des commutateurs s'ils le permettent (fonction *Port Security*). Les commutateurs de niveau 3 par exemple offrent la possibilité de paramétrer des associations port/MAC/IP statiques. Mais cela rend évidemment plus difficile la maintenance du parc ;
- de surveiller les messages ARP circulant sur réseau informatique, à l'aide d'outils de surveillance tels qu'ARPwatch[2] ou arpalert[3] ou de systèmes de Détection d'Intrusion (IDS).

Chaque entrée dans la table ARP a une durée de vie, ce qui oblige l'attaquant à corrompre régulièrement le cache de la victime. Certains systèmes d'exploitation comme Solaris permettent de modifier la valeur de ce temps d'expiration (commande `ndd`). Une valeur courte rendra la corruption plus facilement visible.

En-tête ARP

Cas général

+	Bits 0 - 7	8 - 15	16 - 31
0	*Hardware type*		*Protocol type*
32	*Hardware Address Length*	*Protocol Address Length*	*Operation*
64	*Sender Hardware Address*		
?	*Sender Protocol Address*		
?	*Target Hardware Address*		
?	*Target Protocol Address*		

avec :

Hardware type (*type de matériel*)[4]

- 01 - Ethernet (10Mb) [JBP]
- 02 - *Experimental Ethernet* (3Mb) [JBP]

Protocol type (*Type de protocole*)

- 0x0800 - IP

Ce champ indique quel est le type de protocole couche 3 (OSI) qui utilise ARP.

Hardware Address Length (*longueur de l'adresse physique*)

- 01 - *Token Ring*
- 06 - Ethernet

Ce champ correspond à la longueur de l'adresse physique. La longueur doit être prise en octets.

Protocol Address Length (*longueur de l'adresse logique*)

- 04 - IP v4
- 16 - IP v6

Ce champ correspond à la longueur de l'adresse réseau. La longueur doit être prise en octets.

Operation

- 01 - *Request requête*

- 02 - *Reply réponse*

Ce champ permet de connaître la fonction du message et donc son objectif.

Sender Hardware Address (*adresse physique de l'émetteur*)

Adresse MAC source dans le cadre d'Ethernet.

Sender Internet Address (*adresse réseau de l'émetteur*)

Adresse IP de source dans le cadre de TCP/IP.

Target Hardware Address (*adresse physique du destinataire*)

Adresse MAC destination dans le cadre d'Ethernet. Si c'est une demande ARP, alors, ne connaissant justement pas cette adresse, le champ sera mis à 0.

Target Internet Address (*adresse réseau du destinataire*)

Adresse IP de destination dans le cadre de TCP/IP

Octet 1	**Octet 2**	**Octet 3**	**Octet 4**
0x0001		0x0800	
0x06	0x04	*Operation*	
Adresse MAC source (octets 1-4)			
Adresse MAC source (octets 5-6)		Adresse IP source (octets 1-2)	
Adresse IP source (octets 3-4)		Adresse MAC destination (octets 1-2)	
Adresse MAC destination (octets 3-6)			
Adresse IP destination (octets 1-4)			

|+ Exemple d'en-tête ARP : protocole IPv4 sur Ethernet (28 octets)

Requêtes ARP gratuites

Des requêtes ARP gratuites (*gratuitous ARP*) sont envoyées au démarrage de certains systèmes d'exploitation. Par exemple, certains modems-routeurs envoient ce type de requêtes au démarrage. Elles permettent à cet équipement, nouvel arrivant sur le réseau, de vérifier que son adresse IP n'existe pas déjà, ce qui évite des conflits par doublon d'adresse IP[5]. L'interface expéditeur de la requête n'attend aucune réponse. La mise à jour de la mémoire tampon des systèmes connectés au réseau est alors assurée. Les commutateurs sont informés de l'existence de l'adresse MAC de la machine en question. L'ensemble de ces actions assure une plus grande rapidité ultérieure de connexion au réseau. Une multitude d'émission de ces types de requêtes peut être un indicateur de câble défectueux entrainant des reconnexions fréquentes[6].

Notes et références

[1] http://fr.wikipedia.org/w/index.php?title=Address_Resolution_Protocol&action=edit&section=0
[2] Outil du (NRG), (ICSD), (LBNL) (http://www-nrg.ee.lbl.gov/).
[3] arpalaert (http://www.arpalert.net/).
[4] (http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml#hardware-type-rules), sur le site de l'IANA.
[5] Notons que la RFC 3927 (http://www.ietf.org/rfc/rfc3927.txt?number=3927) déconseille des envois périodiques de requêtes ARP gratuites.
[6] *Gratuitous ARP* selon WireShark (http://wiki.wireshark.org/Gratuitous_ARP)

Liens externes

- (en) RFC 826 *An Ethernet Address Resolution Protocol*
- (en) RFC 2390 *Inverse Address Resolution Protocol*
- (en) RFC 5494 *IANA Allocation Guidelines for the Address Resolution Protocol (ARP)*
- (en) ARP Parameters (http://www.iana.org/assignments/arp-parameters/arp-parameters.xhtml) (IANA)
- (fr) Jouer avec le protocole ARP (http://sid.rstack.org/arp-sk/article/arp.html)
- (fr) ARP sur FrameIP (http://www.frameip.com/entete-arp/)
- (fr) Outil gratuit permettant de générer des datagrammes ARP (http://www.authsecu.com/arpflood/) Code source en C fourni

- Portail de l'informatique
- Portail des télécommunications

IPv4

Pile de protocoles
7. Application
6. Présentation
5. Session
4. Transport
3. Réseau
2. Liaison
1. Physique
Modèle Internet
Modèle OSI
modifier [1]

IPv4 (**Internet Protocol version 4)** est la première version d'Internet Protocol (IP) à avoir été largement déployée, et qui forme encore en 2013 la base de la majorité des communications sur Internet, avec l'IPv6. Elle est décrite dans la RFC 791 de septembre 1981, remplaçant la RFC 760, définie en janvier 1980.

Chaque interface d'un hôte IPv4 se voit attribuer une ou plusieurs adresses IP codées sur 32 bits. Au maximum 4 294 967 296 (soit 2^{32}) adresses peuvent donc être attribuées simultanément en théorie (en pratique, un certain nombre ne sont pas utilisables).

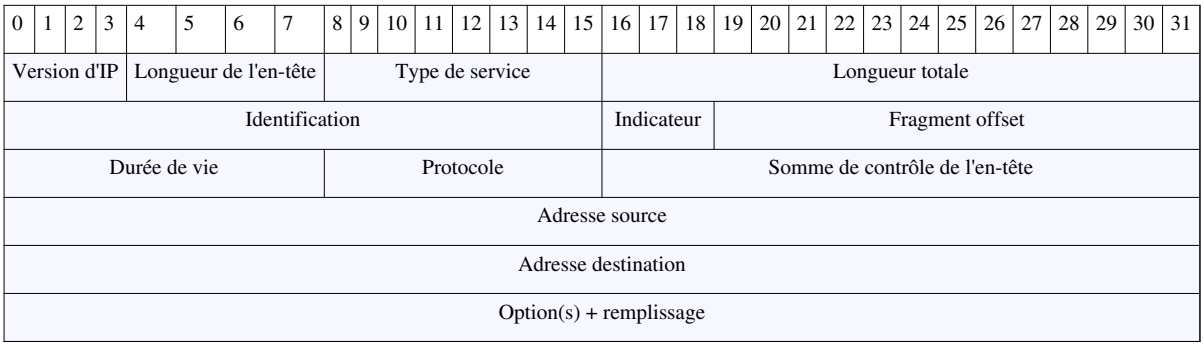
L'épuisement des adresses IPv4 a conduit au développement d'une nouvelle version d'IP, IPv6, et à la transition d'IPv4 vers IPv6 afin d'adopter cette nouvelle version. Le manque d'adresse IPv4 est dans un premier temps contourné grâce à l'utilisation de techniques de traduction d'adresses (NAT) ainsi que par l'adoption du système CIDR. Le nombre d'adresses IP Version 4 publiques est arrivé officiellement à saturation le 3 février 2011.

Représentation d'une adresse IPv4

Une adresse IPv4 est représentée sous la forme de quatre nombres décimaux séparés par des points comme par exemple `193.43.55.67`. Chacun des nombres représente un octet. Pour généraliser, la plage attribuable va donc de 0.0.0.0 à 255.255.255.255 si on fait abstraction des contraintes techniques du protocole (adresse réservée, attribuée...).

En-tête IPv4

En-tête IPv4



Version (4 bits) :

version d'IP utilisée. Ici, 4.

Longueur de l'en-tête ou *IHL* (pour *Internet Header Length*) (4 bits) :

Nombre de mots de 32 bits, soit 4 octets (ou nombre de lignes du schéma). La valeur est comprise entre 5 et 15, car il y a 20 octets minimum et on ne peut dépasser 40 octets d'option (soit en tout, 60 octets).

Type de service ou *ToS* (pour *Type of Service*) (8 bits) :

Ce champ permet de distinguer différentes qualité de service différenciant la manière dont les paquets sont traités. Composé de 3 bits de priorité (donc 8 niveaux) et trois indicateurs permettant de différencier le débit, le délai ou la fiabilité.

Il se décline au fil des RFC. Au départ (RFC 791) nous avions :

```
Bits 0-2:  Precedence.
Bit    3:  0 = Normal Delay,      1 = Low Delay.
Bits   4:  0 = Normal Throughput, 1 = High Throughput.
Bits   5:  0 = Normal Reliability, 1 = High Reliability.
Bit  6-7:  Reserved for Future Use.
```

Le champ DSCP généralise ensuite le champ TOS (RFC 2474, décembre 1998) :

```
Bits 0-5:   DSCP (Differentiated Services Code Point)
Bits 6-7:   CU (Currently Unused)
```

En septembre 2001 (RFC 3168) le champ CU est utilisé pour la gestion des congestions[2] :

```
Bits 0-5:   DSCP (Differentiated Services Code Point)
Bits 6-7:   ECN (Explicit Congestion Notification)
```

Longueur totale en octets ou *Total Length* (16 bits) :

Nombre total d'octets du datagramme, en-tête IP comprise. Donc, la valeur maximale est (2^{16})-1 octets.

Identification (16 bits) :

Numéro permettant d'identifier les fragments d'un même paquet.

Indicateurs ou *Flags* (3 bits) :

1. (Premier bit) actuellement inutilisé.
2. (Deuxième bit) DF (*Don't Fragment*) : lorsque ce bit est positionné à 1, il indique que le paquet ne peut pas être fragmenté. Si le routeur ne peut acheminer ce paquet (taille du paquet supérieure à la MTU), il est alors rejeté.
3. (Troisième bit) MF (*More Fragments*) : quand ce bit est positionné à 1, on sait que ce paquet est un fragment de données et que d'autres doivent suivre. Quand il est à 0, soit le fragment est le dernier, soit le paquet n'a pas été fragmenté.

Fragment offset (13 bits) :

Position du fragment par rapport au paquet de départ, en nombre de mots de 8 octets.

Durée de vie ou TTL (pour *Time To Live*) (8 bits) :

Initialisé par l'émetteur, ce champ est décrémenté d'une unité généralement à chaque saut de routeur. Quand TTL = 0, le paquet est abandonné et un message ICMP est envoyé à l'émetteur pour information.

Protocole (8 bits) :

numéro du protocole au-dessus de la couche réseau : TCP = 6, UDP = 17, ICMP = 1.

Ce champ permet d'identifier le protocole utilisé par le niveau supérieur :

- Internet Control Message Protocol ou ICMP est repéré par les bits 00000001, qu'on écrit souvent en hexadécimal avec 01
- Transmission Control Protocol ou TCP par les bits 00000110, soit 06
- User Datagram Protocol ou UDP par les bits 00010001, soit 17 en décimal

Somme de contrôle de l'en-tête ou *Header Checksum* (16 bits) :

Complément à un de la somme complémentée à un de tout le contenu de l'en-tête afin de détecter les erreurs de transfert. Si la somme de contrôle est invalide, le paquet est abandonné sans message d'erreur.

Adresse source (32 bits) :

Adresse IP de l'émetteur sur 32 bits.

Adresse destination (32 bits) :

Adresse IP du récepteur 32 bits.

Options (0 à 40 octets par mots de 4 octets) :

Facultatif.

Remplissage ou *Padding* :

Champ de taille variable comprise entre 0 et 7 bits. Il permet de combler le champ option afin d'obtenir un en-tête IP multiple de 32 bits. La valeur des bits de bourrage est 0.

Fragmentation

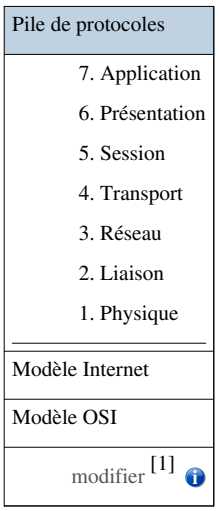
Sur une interface déterminée, une trame a une taille maximale, appelée *Maximum Transmission Unit* ou MTU. Lorsque la longueur du paquet (datagramme) est supérieure, l'information sera fragmentée. La taille maximum supportée par IPv4 (car codée sur 16 bits) est de 64 Ko mais les réseaux ne prennent généralement pas en charge de trames de telles longueurs, en général on trouve des MTU de l'ordre de 1 500 octets (Ethernet).

Notes et références

[1] http://fr.wikipedia.org/w/index.php?title=IPv4&action=edit&section=0
[2] cf. registres DSCP par l'IANA (http://www.iana.org/assignments/dscp-registry)

User Datagram Protocol

Pour les articles homonymes, voir UDP.



Le ***User Datagram Protocol*** (**UDP**, en français **protocole de datagramme utilisateur**) est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport de la pile de protocole TCP/IP : dans l'adaptation approximative de cette dernière au modèle OSI, il appartiendrait à la couche 4, comme TCP. Il est détaillé dans la RFC 768.

Le rôle de ce protocole est de permettre la transmission de données de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Contrairement au protocole TCP, il fonctionne sans négociation : il n'existe pas de procédure de connexion préalable à l'envoi des données (le handshaking). Donc UDP ne garantit pas la bonne livraison des datagrammes à destination, ni leur ordre d'arrivée. Il est également possible que des datagrammes soient reçus en plusieurs exemplaires.

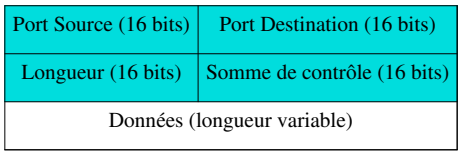
L'intégrité des données est assurée par une somme de contrôle sur l'en-tête. L'utilisation de cette somme est cependant facultative en IPv4 mais obligatoire avec IPv6. Si un hôte n'a pas calculé la somme de contrôle d'un datagramme émis, la valeur de celle-ci est fixée à zéro. La somme de contrôle inclut les adresses IP source et destination.

La nature de UDP le rend utile pour transmettre rapidement de petites quantités de données, depuis un serveur vers de nombreux clients ou bien dans des cas où la perte d'un datagramme est moins gênante que l'attente de sa retransmission. Le DNS, la voix sur IP ou les jeux en ligne sont des utilisateurs typiques de ce protocole.

Structure d'un datagramme UDP

Le paquet UDP est encapsulé dans un paquet IP. Il comporte un en-tête suivi des données proprement dites à transporter.

L'en-tête d'un datagramme UDP est plus simple que celui de TCP :



Il contient les quatre champs suivants :

Port Source

indique depuis quel port le paquet a été envoyé.

Port de Destination

indique à quel port le paquet doit être envoyé.

Longueur

indique la longueur totale (exprimée en octets) du segment UDP (en-tête et données). La longueur minimale est donc de 8 octets (taille de l'en-tête).

Somme de contrôle

celle-ci permet de s'assurer de l'intégrité du paquet reçu quand elle est différente de zéro. Elle est calculée sur l'ensemble de l'en-tête UDP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)
Note : la présence de ce pseudo en-tête, interaction entre les deux couches IP et UDP, est une des raisons qui font que le modèle TCP/IP ne s'applique pas parfaitement au modèle OSI.

La table suivante décrit les champs utilisés pour le calcul de la somme de contrôle UDP sur IPv4 (les indices négatifs correspondent au pseudo en-tête IP) :

+	Bits 0 - 7	8 - 15	16 - 23	24 - 31
-96	Adresse Source			
-64	Adresse Destination			
-32	Zéros	Protocole	Taille UDP	
0	Port Source		Port Destination	
32	Taille		Checksum	
64	Data			

Utilisation

Il est utilisé quand il est nécessaire soit de transmettre des données très rapidement, et où la perte d'une partie de ces données n'a pas grande importance, soit de transmettre des petites quantités de données, là où la connexion « *3-WAY* » TCP serait inutilement coûteuse en ressources. Par exemple, dans le cas de la transmission de la voix sur IP, la perte occasionnelle d'un paquet est tolérable dans la mesure où il existe des mécanismes de substitution des données manquantes, par contre la rapidité de transmission est un critère primordial pour la qualité d'écoute.

Exemples d'utilisation :

- les protocoles DNS, SNMP, TFTP ;

- le *streaming* ;
- les jeux en réseau (exemple : jeux de tir subjectifs) ;
- le programme traceroute.

Liens externes

- (en) RFC 768 – *User Datagram Protocol*
- (en) *IANA Port Assignments* [2] – liste des ports prédéfinis et de leurs utilisations, par l'IANA
- Portail des télécommunications

Références

[1] http://fr.wikipedia.org/w/index.php?title=User_Datagram_Protocol&action=edit&section=0
[2] http://www.iana.org/assignments/port-numbers

Transmission Control Protocol

Pour les articles homonymes, voir TCP.

Pile de protocoles
7. Application 6. Présentation 5. Session 4. Transport 3. Réseau 2. Liaison 1. Physique
Modèle Internet
Modèle OSI
modifier [1]

Transmission Control Protocol (littéralement, « protocole de contrôle de transmissions »), abrégé **TCP**, est un protocole de transport fiable, en mode connecté, documenté dans la RFC 793[2] de l'IETF.

Dans le modèle Internet, aussi appelé modèle TCP/IP, TCP est situé au-dessus de IP. Dans le modèle OSI, il correspond à la couche transport, intermédiaire de la couche réseau et de la couche session. Les applications transmettent des flux de données sur une connexion réseau. TCP découpe le flux d'octets en *segments* dont la taille dépend de la MTU du réseau sous-jacent (couche liaison de données).

TCP a été développé en 1973 puis adopté pour Arpanet en 1983, remplaçant NCP (RFC 801[3]).

Fonctionnement

Une session TCP fonctionne en trois phases :

- l'établissement de la connexion ;
- les transferts de données ;
- la fin de la connexion.

L'établissement de la connexion se fait par un handshaking en trois temps. La rupture de connexion, elle, utilise un handshaking en quatre temps. Pendant la phase d'établissement de la connexion, des paramètres comme le numéro de séquence sont initialisés afin d'assurer la transmission fiable (sans perte et dans l'ordre) des données.

Structure d'un segment TCP

En bits

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td colspan="16">Port Source 2 octets</td><td colspan="16">Port destination 2 octets</td></tr>
<tr><td colspan="32">Numéro de séquence</td></tr>
<tr><td colspan="32">Numéro d'acquittement</td></tr>
<tr><td colspan="4">Taille de l'en-tête</td><td colspan="6">Réservé</td><td>ECN</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Fenêtre</td></tr>
<tr><td colspan="16">Somme de contrôle</td><td colspan="16">Pointeur de données urgentes</td></tr>
<tr><td colspan="24">Options</td><td colspan="8">Remplissage</td></tr>
<tr><td colspan="32">Données</td></tr>
</table>

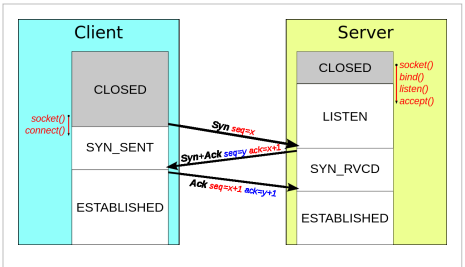
Signification des champs :

- Port source : numéro du port source
- Port destination : numéro du port destination
- Numéro de séquence : numéro de séquence du premier octet de ce segment
- Numéro d'acquittement : numéro de séquence du prochain octet attendu
- Taille de l'en-tête : longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête)
- Drapeaux
  - Réservé : réservé pour un usage futur
  - ECN : signale la présence de congestion, voir RFC 3168
  - URG : Signale la présence de données **urg**entes
  - ACK : signale que le paquet est un accusé de réception (**ack**nowledgement)
  - PSH : données à envoyer tout de suite (**push**)
  - RST : rupture anormale de la connexion (**rese**t)
  - SYN : demande de synchronisation (SYN) ou établissement de connexion
  - FIN : demande la FIN de la connexion
- Fenêtre : taille de fenêtre demandée, c'est-à-dire le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- Somme de contrôle : somme de contrôle calculée sur l'ensemble de l'en-tête TCP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)
- Pointeur de données urgentes : position relative des dernières données urgentes
- Options : facultatives
- Remplissage : zéros ajoutés pour aligner les champs suivants du paquet sur 32 bits, si nécessaire

- Données : séquences d'octets transmis par l'application (par exemple : +OK POP3 server ready...)

Établissement d'une connexion

Article détaillé : Three-way handshake.



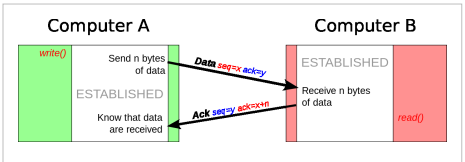
Même s'il est possible pour deux systèmes d'établir une connexion entre eux simultanément, dans le cas général, un système ouvre une 'socket' (point d'accès à une connexion TCP) et se met en attente passive de demandes de connexion d'un autre système. Ce fonctionnement est communément appelé *ouverture passive*, et est utilisé par le côté *serveur* de la connexion. Le côté *client* de la connexion effectue une *ouverture active* en 3 temps :

1. Le client envoie un segment SYN au serveur,
2. Le serveur lui répond par un segment SYN/ACK,
3. Le client confirme par un segment ACK.

Durant cet échange initial, les numéros de séquence des deux parties sont synchronisés :

1. Le client utilise son numéro de séquence initial dans le champ "Numéro de séquence" du segment SYN (x par exemple),
2. Le serveur utilise son numéro de séquence initial dans le champ "Numéro de séquence" du segment SYN/ACK (y par exemple) et ajoute le numéro de séquence du client plus un (x+1) dans le champ "Numéro d'acquittement" du segment,
3. Le client confirme en envoyant un ACK avec un numéro de séquence augmenté de un (x+1) et un numéro d'acquittement correspondant au numéro de séquence du serveur plus un (y+1).

Transferts de données



Pendant la phase de transferts de données, certains mécanismes clefs permettent d'assurer la robustesse et la fiabilité de TCP. En particulier, les numéros de séquence sont utilisés afin d'ordonner les segments TCP reçus et de détecter les données perdues, les sommes de contrôle permettent la détection d'erreurs, et les acquittements ainsi que les temporisations permettent la détection des segments perdus ou retardés.

Numéros de séquence et d'acquittement

Grâce aux numéros de séquence et d'acquittement, les systèmes terminaux peuvent remettre les données reçues dans l'ordre à l'application destinataire.

Les numéros de séquence sont utilisés pour décompter les données dans le flux d'octets. On trouve toujours deux de ces nombres dans chaque segment TCP, qui sont le *numéro de séquence* et le *numéro d'acquittement*. Le *numéro de séquence* représente le propre numéro de séquence de l'émetteur TCP, tandis que le *numéro d'acquittement* représente le numéro de séquence du destinataire. Afin d'assurer la fiabilité de TCP, le destinataire doit acquitter les segments reçus en indiquant qu'il a reçu toutes les données du flux d'octets jusqu'à un certain numéro de séquence.

Le numéro de séquence indique le premier octet des données.

Par exemple, dans le cas d'un échange de segments par Telnet :

1. L'hôte A envoie un segment à l'hôte B contenant un octet de données, un *numéro de séquence* égal à 42 (Seq = 42) et un *numéro d'acquittement* égal à 79 (Ack = 79),
2. L'hôte B envoie un segment ACK à l'hôte A. Le *numéro de séquence* de ce segment correspond au *numéro d'acquittement* de l'hôte A (Seq = 79) et le *numéro d'acquittement* au *numéro de séquence* de A tel que reçu par B,

augmenté de la quantité de données en bytes reçue (Ack = 42 + 1 = 43),

3. L'hôte A confirme la réception du segment en envoyant un ACK à l'hôte B, avec comme *numéro de séquence* son nouveau *numéro de séquence*, à savoir 43 (Seq = 43) et comme *numéro d'acquittement* le *numéro de séquence* du segment précédemment reçu, augmenté de la quantité de données reçue (Ack = 79 + 1 = 80).

Les numéros de séquence sont des nombres entiers non signés sur 32 bits, qui reviennent à zéro après avoir atteint 2^32-1. Le choix du numéro de séquence initial est une des clefs de la robustesse et de la sécurité des connexions TCP.

Une amélioration de TCP, nommée acquittement sélectif (*selective acknowledgement* ou SACK), autorise le destinataire TCP à acquitter des blocs de données reçus dans le désordre.

Somme de contrôle

Une somme de contrôle sur 16 bits, constituée par le complément à un de la somme complémentée à un de tous les éléments d'un segment TCP (en-tête et données), est calculée par l'émetteur, et incluse dans le segment émis. Le destinataire recalcule la somme de contrôle du segment reçu, et si elle correspond à la somme de contrôle reçue, on considère que le segment a été reçu intact et sans erreur.

Temporisation

La perte d'un segment est gérée par TCP en utilisant un mécanisme de temporisation et de retransmission. Après l'envoi d'un segment, TCP va attendre un certain temps la réception du ACK correspondant. Un temps trop court entraîne un grand nombre de retransmissions inutiles et un temps trop long ralentit la réaction en cas de perte d'un segment.

Dans les faits, le délai avant retransmission doit être supérieur au RTT moyen d'un segment, c'est-à-dire au temps que prend un segment pour effectuer l'aller-retour entre le client et le serveur. Comme cette valeur peut varier dans le temps, on "prélève" des échantillons à intervalle régulier et on en calcule une moyenne pondérée :

```
RTT moyen = (1-α) * RTT moyen + α * RTT échantillon
```

Une valeur typique pour α est 0.125. L'influence des échantillons diminue de manière exponentielle dans le temps.

Le délai à utiliser est obtenu à partir de cette estimation du RTT moyen et en y ajoutant une marge de sécurité. Plus la différence entre un échantillon et la moyenne est grande, plus la marge de sécurité à prévoir est importante. Le calcul se fait à partir de la variance pondérée entre l'échantillon et la moyenne :

```
Variance RTT = (1-β) * Variance RTT + β * |RTT échantillon - RTT moyen|
```

Une valeur typique pour β est 0.25. Le délai à utiliser est finalement donné par la formule suivante :

```
Délai = RTT moyen + 4 * Variance RTT
```

L'Algorithme de Karn permet de mieux évaluer le délai en présence d'*acquittements ambigus*. En effet, si un segment envoyé a été perdu, les segments ultérieurs provoqueront des acquittements où figurera le numéro du premier octet manquant, et on ne sait donc plus à quel segment envoyé correspondent ces acquittements.

Parfois, quand le délai est trop long, il est avantageux de ne pas attendre avant de retransmettre un segment. Si un hôte reçoit 3 ACKs pour le même segment, alors il considère que tous les segments transmis après le segment acquitté ont été perdus et les retransmet donc immédiatement (*Fast retransmit*).

Contrôle de flux

Chaque partenaire dans une connexion TCP dispose d'un tampon de réception dont la taille n'est pas illimitée. Afin d'éviter qu'un hôte ne surcharge l'autre, TCP prévoit plusieurs mécanismes de contrôle de flux. Ainsi, chaque segment TCP contient la taille disponible dans le tampon de réception de l'hôte qui l'a envoyé. En réponse, l'hôte distant va limiter la taille de la fenêtre d'envoi afin de ne pas le surcharger.

D'autres algorithmes comme Nagle ou Clarck facilitent également le contrôle du flux.

Contrôle de congestion

La congestion intervient lorsque trop de sources tentent d'envoyer trop de données trop vite pour que le réseau soit capable de les transmettre. Ceci entraîne la perte de nombreux paquets et de longs délais.

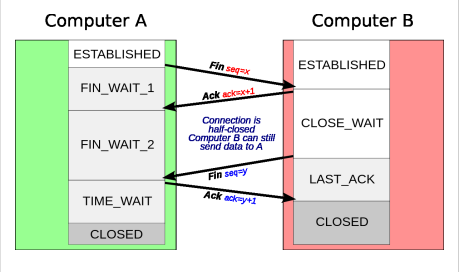
Les acquittements des données émises, ou l'absence d'acquittements, sont utilisés par les émetteurs pour interpréter de façon implicite l'état du réseau entre les systèmes finaux. À l'aide de temporisations, les émetteurs et destinataires TCP peuvent modifier le comportement du flux de données. C'est ce qu'on appelle généralement le contrôle de congestion.

Il existe une multitude d'algorithmes d'évitement de congestion pour TCP.

Autres

TCP utilise un certain nombre de mécanismes afin d'obtenir une bonne robustesse et des performances élevées. Ces mécanismes comprennent l'utilisation d'une fenêtre glissante, l'algorithme de démarrage lent (*slow start*), l'algorithme d'évitement de congestion (*congestion avoidance*), les algorithmes de retransmission rapide (*fast retransmit*) et de récupération rapide (*fast recovery*), etc. Des recherches sont menées actuellement afin d'améliorer TCP pour traiter efficacement les pertes, minimiser les erreurs, gérer la congestion et être rapide dans des environnements très haut débit.

Terminaison d'une connexion



La phase de terminaison d'une connexion utilise un handshaking en quatre temps, chaque extrémité de la connexion effectuant sa terminaison de manière indépendante. Ainsi, la fin d'une connexion nécessite une paire de segments FIN et ACK pour chaque extrémité.

Ports TCP

TCP, comme UDP, utilise le numéro de port pour identifier les applications. À chaque extrémité (client/serveur) de la connexion TCP est associé un numéro de port sur 16 bits (de 1 à 65535) assigné à l'application émettrice ou réceptrice. Ces ports sont classés en trois catégories :

- Les *ports bien connus* sont assignés par l'IANA (Internet Assigned Numbers Authority) dans la plage 0-1023, et sont souvent utilisés par des processus système ou ayant des droits privilégiés. Les applications bien connues qui fonctionnent en tant que serveur et sont en attente de connexions utilisent généralement ces types de ports. Exemples : FTP (21), SSH (22), Telnet (23), SMTP (25), HTTP (80), POP3 (110).
- Les *ports enregistrés* sont généralement utilisés par des applications utilisateur comme ports sources éphémères pour se connecter à un serveur, mais ils peuvent aussi identifier des services non enregistrés par l'IANA.
- Les *ports dynamiques/privés* peuvent aussi être utilisés par des applications utilisateur, mais plus rarement. Ils n'ont pas de sens en dehors d'une connexion TCP particulière.

Développement de TCP

C'est le ministère américain de la Défense (DoD) qui à l'origine a développé le modèle de référence TCP/IP, car il avait besoin d'un réseau pouvant résister à toutes les situations.

TCP est un protocole assez complexe, et en évolution. Même si des améliorations significatives ont été apportées au cours des années, son fonctionnement de base a peu changé depuis le RFC 793 [4], publié en 1981. Le RFC 1122 [5] (*Host Requirements for Internet Hosts*), a clarifié un certain nombre de pré-requis pour l'implémentation du protocole TCP. Le RFC 2581 [6] (*TCP Congestion Control*), l'un des plus importants de ces dernières années, décrit de nouveaux algorithmes utilisés par TCP pour éviter les congestions. En 2001, le RFC 3168 [7] a été écrit afin de présenter un mécanisme de signalisation des congestions (*explicit congestion notification* ou ECN), et s'ajoute à la liste des RFC importants qui complètent la spécification originale. Au début du XXI^{e} siècle, TCP est utilisé approximativement pour 95 % de tout le trafic Internet. Les applications les plus courantes qui utilisent TCP sont HTTP/HTTPS (World Wide Web), SMTP/POP3/IMAP (messagerie) et FTP (transfert de fichiers).

Alternatives à TCP

De nombreuses applications en temps réel n'ont pas besoin, et peuvent même souffrir, des mécanismes complexes de transport fiable de TCP : applications de diffusion multimédia (audio, vidéo), certains jeux multi-joueurs en temps réel, échanges de fichiers, etc. Dans ce type d'applications, il est souvent préférable de gérer les pertes, erreurs ou congestions, plutôt que d'essayer de les éviter.

Pour ces besoins particuliers, d'autres protocoles de transport ont été créés et déployés.

- UDP (User datagram protocol) est souvent utilisé lorsque le temps-réel est privilégié sur la fiabilité. Tout comme TCP, ce protocole propose un multiplexage applicatif à travers la notion de ports, mais fonctionne en mode non connecté.
- SCTP (Stream Control Transmission Protocol), protocole fournissant des services similaires à TCP (fiabilité, remise en ordre des séquences, et contrôle de congestion), tout en offrant la possibilité de communications multi-cibles comme avec UDP.
- MPTCP a pour but de permettre une même connexion TCP à travers différentes interfaces réseau, par exemple passer des réseaux GSM 3G/4G au Wi-Fi.

Références

[1] http://fr.wikipedia.org/w/index.php?title=Transmission_Control_Protocol&action=edit&section=0
[2] (en) « TRANSMISSION CONTROL PROTOCOL (http://tools.ietf.org/html/rfc793) », Request for comments n° 793, Septembre 1981>.
[3] (en) « NCP/TCP TRANSITION PLAN (http://tools.ietf.org/html/rfc801) », Request for comments n° 801, novembre 1981.
[4] http://tools.ietf.org/html/rfc793
[5] http://tools.ietf.org/html/rfc1122
[6] http://tools.ietf.org/html/rfc2581
[7] http://tools.ietf.org/html/rfc3168

Sources et contributeurs de l'article

Ethernet *Source*: http://fr.wikipedia.org/w/index.php?oldid=107305172 *Contributeurs*: -Nmd, Abrahami, Archimëa, Arno., Artificis, B3nZ3n, Bac's, BernardM, Bertrand Fr 24, Bobblewik, Bobodu63, Briling, Brohee, Bub's, Buzz, Calo, Capbat, Captainm, Cburnett, Chaouistes, ChrisJ, Christyo, Claratte, Cnuma, Coffrini, CommonsDelinker, Coyote du 86, Cyrienna, DainDwarf, David Berardan, EDUCA33E, Eberkut, Eric german, Escaladix, Eusebius, Fedmahn, Feldo, Freewol, Frodary, GLec, GenEars, Ggal, Gédé, Hibou57, Ima-polytech-lille, Inike, Iznogood, JackPotte, Jerome66, Jmax, Keikomi, Kilith, Koko90, Kropotkine 113, Le G.O., Le pro du 94 :), LeFabz, Leag, Lenaic, Letartean, Litlok, Loader, Looxix, LyonL, MagnetiK, Manu1400, Maurice Akin, Mcannac, Med, MetalGearLiquid, Michbeie, Mikayé, Mike-m, Mro, Mudares, Nathan30, NicoRay, Nykozoft, Olrick, Orthogaffe, Oz, P-e, Pabix, Pano38, PatLeNain, Patatosaure, Pautard, PierreSelim, Pixeltoo, Ploum's, Plyd, Popolon, R, Raph, RaphAstronome, Roffetn2, Romainb, Romainhk, Romanc19s, Ryo, Rémih, Sam Hocevar, Sbrunner, Seafire, Sebf, Smainlak, SteF, Stephane.lecorne, SteveZodiac, Tango Panaché, Tbowan, Tifrere, Tiro, Trantor, Triba, Vargenau, Verdy p, W'rkncacnter, Xfigpower, Xofc, Xr, Yann Lejeune, Zakary6666666666666666, Zardoz69, marseille-4-a7-62-147-114-146.dial.proxad.net, script de conversion, 189 modifications anonymes

Address Resolution Protocol *Source*: http://fr.wikipedia.org/w/index.php?oldid=104498356 *Contributeurs*: -Nmd, AFAccord, Acp, Aoineko, Bob08, Cakoin, CaptainHaddock, Chevalier libre, Crochet.david, David Berardan, Domingue, EDUCA33E, Elfi, Eric german, Francois Trazzi, François-Dominique, GLec, Garandel, Gdgourou, H. fadili, Hamza ab, Houssem bahri, Inike, Jtchaneg, JujuTh, Le gorille, Lomita, Manouille, MathsPoetry, Med, MetalGearLiquid, Moala, Mro, NaSH, Nico45, Nono64, Olbat, Orthogaffe, Oz, P1gu1n, Phe, Phetu, Philippe97, Pspathis, R, Rangzen, Romanc19s, Rémih, Sebf, SlyWax, Spooky, SuDForcE, Symac, T, Topeil, Treanna, Turb, Vladoulianov, Wcorrector, 52 modifications anonymes

IPv4 *Source*: http://fr.wikipedia.org/w/index.php?oldid=102828164 *Contributeurs*: -Nmd, Abdo1985dj, Akiry, B.bellec, BTCK, Bob08, Boretti, Bub's, David Berardan, Djo0012, Eberkut, Emirix, Eric german, Eusebius, Floflo, Francois Trazzi, Freewol, G-37, G.lardoux, GLec, Gibux, Gotrek, Haugure, Houston83, Hégésippe Cormier, Ilario, Jef-Infojef, Julienjig, Kevin.mille, Laocian, Liquid 2003, Lomita, MaTT, Marc Mongenet, MetalGearLiquid, Mike2, Mro, Pascal VILLARS, Poil, R, Redox, Rohanec, Sebf, Shawn, Sherbrooke, Shinfacts, Ssx`z, Toshiro92, Ulysse2000, Wanderer999, Zecko, 59 modifications anonymes

User Datagram Protocol *Source*: http://fr.wikipedia.org/w/index.php?oldid=105720707 *Contributeurs*: -Nmd, Alvaro, Arno., Athymik, BZP, Bap, Bikepunk2, Bub's, Djmoa, Eberkut, Echoray, Eusebius, Fab97, Francois Trazzi, GLec, Guillaume Pousse, Hémant, JMVF, Jon207, Lthevenet, Melkor73, MetalGearLiquid, Mro, Nono64, Orthogaffe, Oz, PouX, R, Romainhk, Ryo, Sebf, SniperMaské, Vonvon, Wcorrector, Webkid, 44 modifications anonymes

Transmission Control Protocol *Source*: http://fr.wikipedia.org/w/index.php?oldid=106010165 *Contributeurs*: -Nmd, Alvaro, Andre Engels, Aoineko, Athymik, BZP, Benjamin, Bikepunk2, BlaF, Bub's, Buzz, Calo, Canarix, Capbat, ChrisJ, Daba, David Berardan, Davitof, Denis Dordoigne, Eberkut, Elfix, Erdnaxeli, Etudiant Metz, Eusebius, Fab97, Francois Trazzi, Fylip22, GLec, Hemmer, IRedRat, JMVF, JackPotte, Jejelefou, Jerikojerk, Jmax, Jmdavid1789, Jona, Jonathan Despraz, Julien Stuby, Jylam, Koyuki, L0stman, Leag, LeonardoRob0t, Lomita, Luna1253, Lyondif02, Magellan, Marc Mongenet, MathsPoetry, Mektroid, Melkor73, Milord, Moala, Mro, NaSH, Nef, Noar, Nono64, Nykozoft, Olyvar, Orlodrim, Orthogaffe, Oussama817, Oz, Pautard, Pixeltoo, Quaternion, R, Rhizome, Romanc19s, Rune Obash, Ryo, Sam Hocevar, Sebastienadam, Sebf, Sebleouf, Skc, SniperMaské, Symac, Tieno, Tomates Mozzarella, Trusty, Vargenau, Vonvon, Xavier Combelle, Zelda, Zil, script de conversion, 128 modifications anonymes

Source des images, licences et contributeurs

Fichier:Books-aj.svg aj ashton 01.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Books-aj.svg_aj_ashton_01.svg *Licence*: Public Domain *Contributeurs*: Original author: AJ Ashton (on OpenClipArt). Code fixed by verdy_p for XML conformance, and MediaWiki compatibility, using a stricter subset of SVG without the extensions of SVG editors, also cleaned up many unnecessary CSS attributes, or factorized them for faster performance and smaller size. All the variants linked below are based on this image.

Fichier:Question book-4.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Question_book-4.svg *Licence*: GNU Free Documentation License *Contributeurs*: Tkgd2007

Fichier:Ethernet RJ45 connector p1160054.jpg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Ethernet_RJ45_connector_p1160054.jpg *Licence*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributeurs*: User:David.Monniaux

Image:Gtk-dialog-info.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Gtk-dialog-info.svg *Licence*: GNU Lesser General Public License *Contributeurs*: David Vignoni

Fichier:Ethernet Type II Frame format.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Ethernet_Type_II_Frame_format.svg *Licence*: Public Domain *Contributeurs*: Bruceadler, Jodo, MetalGearLiquid, Mikm, Popolon, Renepick, WikipediaMaster, 3 modifications anonymes

Image:Disambig colour.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Disambig_colour.svg *Licence*: Public Domain *Contributeurs*: Bub's

Fichier:Crystal mycomputer.png *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Crystal_mycomputer.png *Licence*: inconnu *Contributeurs*: Dake, Rocket000

Fichier:Crystal_Clear_app_linneighborhood.png *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Crystal_Clear_app_linneighborhood.png *Licence*: GNU Lesser General Public License *Contributeurs*: Everaldo Coelho and YellowIcon

Fichier:Tcp connect.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Tcp_connect.svg *Licence*: Creative Commons Attribution-Sharealike 3.0 *Contributeurs*: User:Skc

Fichier:Tcp talk.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Tcp_talk.svg *Licence*: Creative Commons Attribution-Sharealike 3.0 *Contributeurs*: User:Skc

Fichier:Tcp close.svg *Source*: http://fr.wikipedia.org/w/index.php?title=Fichier:Tcp_close.svg *Licence*: Creative Commons Attribution-Sharealike 3.0 *Contributeurs*: User:Skc

Licence