Protocoles Réseau III

Juliusz Chroboczek

27 septembre 2022

La couche lien utilise le service fourni par la couche physique pour fournir à la couche réseau un service de communication :

- local au lien ;
- par paquets ;
- non-fiable ; et
- ordonné.

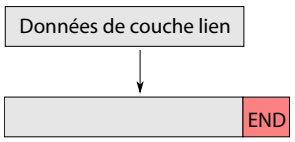
À la couche lien, on appelle les paquets des *trames*.

De même qu'il existe de nombreux protocoles de couche physique, il existe de nombreux protocoles de couche lien. IP(v4) a été conçu pour faire aussi peu de demandes que possible à la couche lien, et fonctionne donc mieux au-dessus d'un protocole de couche lien simple qu'au-dessus d'un protocole complexe : par exemple, si la couche lien essaie de fournir des garanties de qualité de service, IP ne saura pas profiter de ces garanties, et l'effort sera gâché. Cependant, on a dans le passé fait fonctionner avec succès IP au-dessus de couches lien complexes (X.25, ATM), et nous n'avons pas encore totalement guéri des blessures de ces expériences (ADSL).

1 Exemple : le protocole SLIP

Le protocole SLIP (RFC 1055) est un protocole de couche lien simple et élégant qui permet de transférer des paquets IP sur des lignes séries asynchrones. Il a beaucoup été utilisé dans les années 1990 pour fournir un accès par modem à l'Internet.

SLIP est conçu pour fonctionner au-dessus d'une ligne série asynchrone : la couche physique fournit un service de communication par octets, et la couche lien doit les réunir en trames. SLIP utilise pour cela un octet de fin de trame, dénoté par END et valant 192. Pour éviter qu'un octet de valeur 192 soit pris pour une fin de trame, SLIP code chaque 192 par une suite ESC 220, où ESC vaut 219 ; enfin, un octet de valeur 192 est codé par la suite ESC 221 :



- END=192 marque une fin de trame ;
- ESC=219 sert de caractère d'échappement :
  - ESC 220 code 192 ;
  - ESC 221 code 219.

L'émetteur SLIP émet le contenu de la trame en effectuant les deux remplacements ci-dessus, puis envoie un octet END. À la réception, un octet ESC est forcément suivi soit de 220 soit de 221 (sinon, c'est une erreur de *framing*, et la trame est jetée), et un octet END représente forcément une fin de trame ; lorsqu'il rencontre un octet END, le récepteur transmet donc les données décodées à la couche supérieure.

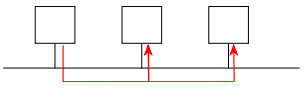
Comparé aux protocoles qui utilisent un codage avec la longueur des données suivie des données elles-même, ce protocole a le désavantage de nécessiter l'échappement des symboles de contrôle, ce qui demande au récepteur de parcourir les données reçues pour inverser l'échappement. Par contre, il a l'énorme avantage d'être *auto-synchronisant (self-synchronising)* : si le récepteur s'est désynchronisé (il a perdu une quantité inconnue de données), il peut se resynchroniser au prochain octet END, qui représente forcément une fin de trame.

Une petite modification à cet algorithme permet de le rendre plus robuste : après une période de silence, l'émetteur envoie un octet END avant d'émettre la trame. Cette modification permet d'éviter que du bruit présent sur la ligne soit concaténé à la première trame après une période de silence.

Digression : le protocole PPP Malgré sa simplicité [1] et son élégance, SLIP n'a pas été adopté par l'IETF : le protocole de couche lien officiel pour les liaisons point-à-point est PPP (RFC 1661).

PPP utilise un *framing* complexe, basé sur le protocole HDLC (obsolète depuis longtemps). Au lieu d'un mécanisme d'échappement, il utilise un codage longueur/données, ce qui permet d'éviter de parcourir les données à envoyer, mais rend la resynchronisation plus difficile. PPP inclut plusieurs champs qui ne s'appliquent pas aux liaisons point-à-point. Son principal avantage est qu'il inclut dans son entête un numéro de protocole de couche supérieure explicite, et permet donc facilement de multiplexer (partager) un lien entre plusieurs protocoles de couche réseau. Il contient aussi un protocole complexe de négociations d'options (omission des champs inutiles, compression d'entêtes, compression des données). PPP permet aussi d'affecter automatiquement des adresses IP — ce qui n'a absolument rien à faire à la couche lien. (IPv6 au-dessus de PPP n'utilise pas cette fonctionnalité, mais utilise un mécanisme d'affectation d'adresses générique de couche réseau.)

2 Exemple : le protocole Ethernet



Ethernet est un protocole à multidiffusion *(broadcast)* originellement conçu pour du câble coaxial « épais » (10Base5). Il a depuis été étendu à de nombreuses autres couches physiques, notamment le coax fin (10Base2), la double paire torsadée (10BaseT), la fibre optique (10BaseFL), et sa vitesse a été augmentée (100BaseTX [2], 1000BaseT etc., jusqu'à 100 Gbit/s). Un format de trame semblable mais incompatible à celui d'Ethernet a été standardisé sous le nom de IEEE 802.2, et le protocole Ethernet lui-même comme IEEE 802.3.

Ethernet utilise une couche physique qui permet de distinguer un signal d'un lien oisif : c'est le *link-sensing*. Par exemple, l'Ethernet à 10 Mbit/s utilise le codage Manchester, et le récepteur

1. Ou du fait de sa simplicité ?
2. Il n'y a pas de 100BaseT — l'Ethernet « rapide » s'appelle officiellement 100BaseTX.

détecte que quelqu'un est en train d'émettre par la présence de transitions.

2.1 Format des trames

Le protocole Ethernet identifie toutes les stations du lien par des adresses de 48 bits (6 octets) dites *adresses MAC*. Une adresse MAC est notée comme 6 nombres hexadécimaux, par exemple `00:11:43:D4:86:A0`.

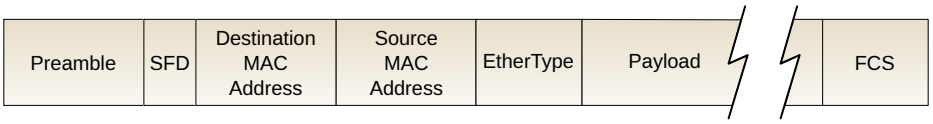


FIGURE 1 — Format d'une trame Ethernet

Une trame Ethernet (Figure 1) est composée d'un *entête Ethernet (Ethernet header)* suivi des données de couche supérieure, elles-mêmes suivies d'un CRC-32. L'entête a la structure suivante :

- le *préambule*, sept octets de 55_{16} qui permettent aux autres stations du lien de synchroniser leur horloge à celle de l'émetteur ;
- un octet valant $5D_{16}$, qui indique le début de la trame ;
- l'adresse MAC du destinataire ;
- l'adresse MAC de l'émetteur ;
- l'*Ethertype*, un entier de deux octets indiquant le protocole de couche supérieure.

Vous remarquerez qu'Ethernet n'utilise de symbole de début ou de fin de trame et ne code pas non plus la longueur des données : la couche physique permet de détecter le début et la fin de la trame (codage Manchester), le récepteur en déduit la taille des données. Le protocole est donc auto-synchronisant à la couche physique.

Ethernet est un réseau à multidiffusion : toutes les stations connectées au lien décodent toutes les trames, puis consultent le champ destination de l'entête pour déterminer si la trame leur est destinée. Si ce n'est pas le cas, elles rejettent simplement la trame. Une adresse MAC particulière, l'*a*dresse de multidiffusion *(broadcast address)* valant `FF:FF:FF:FF:FF:FF`, indique que la trame est destinée à toutes les stations du lien.

2.2 Sous-couche d'accès au lien

Dans un réseau à multidiffusion, il est nécessaire d'éviter que deux stations parlent en même temps. C'est le rôle de la *sous-couche d'accès au lien* ou *MAC (Media Access Control)*. On appelle *collision* la situation où deux trames sont émises en même temps.

La sous-couche MAC d'Ethernet utilise un algorithme qui s'appelle CSMA/CD *(Carrier Sense Multiple Access with Collision Detection)* qui est complètement distribué (chaque station exécute le même algorithme, il n'y a pas de coordinateur central), mais probabiliste : CSMA/CD ne garantit pas l'absence de collisions, mais essaie d'en limiter le nombre et la sévérité. CSMA/CD est une amélioration d'*Aloha*.

2.2.1 Aloha pur

Le protocole Aloha dit « pur » est très simple : lorsqu'une station veut émettre une trame, elle l'émet, sans se soucier des collisions, et c'est aux couches supérieures de gérer la perte de trames. Aloha pur ne demande aucune fonctionnalité particulière à la couche physique, et il a été inventé pour des liens radio où il n'était pas possible d'écouter pendant l'émission [3].

2.2.2 CSMA/CD

L'algorithme CSMA/CD *(Carrier-Sense Multiple Access with Collision Detection)* est une amélioration d'Aloha. Il dépend d'une couche physique où il est possible de passer rapidement de la réception à la transmission, et où il est possible d'entendre la présence d'un signal *(Carrier-Sense)* en même temps qu'on émet. C'est généralement le cas sur les réseaux filaires, et jamais le cas dans les réseaux sans fil.

Carrier-Sense Une station qui veut émettre écoute d'abord pour s'assurer que personne d'autre n'est en train d'émettre. Si c'est le cas, elle émet une trame ; si ce n'est pas le cas, elle attend un temps aléatoire, puis recommence jusqu'à k fois (où k s'appelle le *facteur de persistance* du protocole — on peut parler de *CSMA k-persistant*). Pour éviter les collisions répétées, le temps d'attente avant émission est tiré de façon aléatoire. Pour éviter les instabilités, la constante de temps est augmentée à chaque réémission successive, soit de façon linéaire, soit de façon exponentielle *(backoff)*.

Ce mécanisme ne garantit pas l'absence de collisions ; cependant, du fait du temps d'attente aléatoire, il rend la probabilité de collisions extrêmement faible, surtout lorsque le traffic est asymétrique.

Détection de collisions Si une collision est détectée durant l'émission d'une trame *(Collision Detection)*, la station émettrice interrompt l'émission et envoie un court signal de brouillage *(jam)*. Ce signal permet de détecter la collision, et cause donc un temps d'attente aléatoire suivi d'une réémission. Ce mécanisme ne diminue pas la probabilité de collision, mais en interrompant l'émission réduit le coût d'une collision lorsqu'elle a lieu.

2.2.3 Parenthèse : CSMA/CA

La famille de normes IEEE 802.11, commercialisée sous le nom *Wifi*, est une famille de protocoles pour réseaux sans-fil à faible portée. CSMA/CD ne s'applique pas directement aux réseaux sans fil — la détection de collisions recquiert de pouvoir écouter en même temps qu'on parle.

CSMA/CA contourne ce problème en demandant au récepteur d'acquitter chaque trame, et reémet chaque trame non-acquittée après un *backoff* exponentiel. Il y a trois cas :

- la trame a été perdue du fait d'une collision ; dans ce cas, CSMA/CA réagit presque exactement comme CSMA/CD (qui obéit à un *backoff* exponentiel en cas de collision, mais interrompt l'émission de la trame de façon prématurée, ce qui n'est pas possible avec CSMA/CD) ;

3. En fait, Aloha fait des réémissions — une trame est réémise si elle est perdue, quelle qu'en soit la cause, collision, interférence radio ou congestion d'un routeur. La structure en couches d'Aloha n'étant pas entièrement claire, je préfère considérer cela comme une fonctionnalité de couche supérieure.

– la trame a été perdue pour une autre raison ; dans ce cas, CSMA/CA reément la trame, ce qui est bien le comportement désiré, mais après un temps d'attente qui n'est pas nécessaire dans ce cas ;
– la trame a bien été reçue, mais l'acquittement a été perdu ; dans ce cas, CSCMA/CA reémet inutilement la trame.

On peut donc voir CSMA/CA comme une variante imprécise de CSMA, qui interprète toute perte de trame comme une collision, mais ne recquiert pas de pouvoir écouter en même temps qu'on parle.

Du fait de l'absence de détection de collisions, une collision est beaucoup plus coûteuse en CSMA/CA qu'en CSMA/CD. Pour cette raison, CSMA/CA impose un temps d'attente aléatoire avant l'émission d'une trame même si le lien est libre, ce qui diminue le nombre de collisions mais augmente le temps d'accès au lien.

RTS/CTS RTS/CTS est un protocole aditionnel optionnellement utilisé par IEEE 802.11.

Lorsqu'elle est configurée pour utiliser CSMA/CA, une station qui désire émettre une trame émet une trame RTS *(Request To Send)* ; le destinataire répond par une trame CTS *(Clear To Send)*. Lorsqu'il reçoit la trame CTS, l'émetteur peut émettre la trame. Les trames RTS et CTS contiennent le nombre de bits contenus dans la trame de données qui doit être transmise.

Lorsqu'une station entend une trame RTS ou CTS, elle doit rester silencieuse pendant le temps nécessaire à la transmission de la trame de données.

RTS/CTS réduit drastiquement la probabilité de collision — une trame RTS ou CTS est minuscule, et donc la probabilité de collision entre deux telles trames est très faible ; comme toutes les stations qui entendent une des trames RTS ou CTS restent silencieuses pendant le temps nécessaire à la transmission de données, une collision pendant la transmission de celle-ci est impossible. De plus, elle résoud partiellement le *hidden station problem* — une station qui entend une trame CTS mais pas la trame RTS correspondante peut éviter une collision avec une station qu'elle n'entend pas. Par contre, sur un réseau peu chargé, l'échange RTS/CTS augmente la latence sans aucun bénéfice, ce qui explique que RTS/CTS est optionnel et n'est généralement pas utilisé.

2.3 Ethernet sur paire torsadée et micro-segmentation

Ethernet sur câble coaxial (10Base5, 10Base2) a une topologie en bus — un seul câble rigide auquel sont attachées toutes les stations. Plusieurs autres technologies de réseau permettaient un câblage plus flexible, à base de *hubs* d'interconnexion (ARCnet, StarLan).

La norme 10Base-T a défini une variante d'Ethernet dont la couche physique utilise du câble à double paire torsadée (« câble téléphonique » — c'est le câble Ethernet que vous connaissez) interconnecté par des *hubs*. 10Base-T a été à la base de plusieurs des normes Ethernet qui ont suivi (100Base-TX et 1000Base-T sont les plus répandues aujourd'hui).

Switched Ethernet Ethernet sur paire torsadée (10BaseT et ses successeurs) interconnecte les liens physiques par des *hubs*, qui fonctionnent entièrement à la couche physique : ils recopient le signal électrique sans l'interpréter. Si les hubs évitent les problèmes électriques dans les gros réseaux, il n'évitent pas les collisions, ce qui empêche de faire de trop gros liens.

Un *switch* est un périphérique d'interconnection de couche 2, qui interprète les trames Ethernet. Lorsqu'une trame arrive sur un port d'un *switch*, elle est interprétée, puis reconstruite sur les ports sortants.

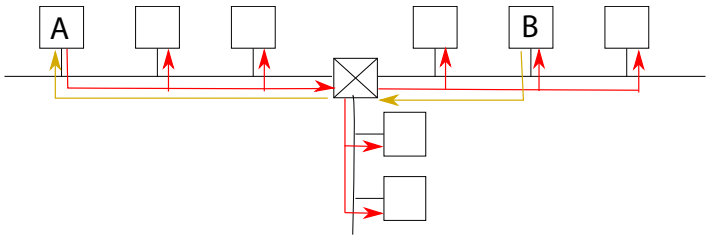


FIGURE 2 — *Opération d'un switch*

Un *switch* implémente une optimisation supplémentaire importante (Figure 2 : il choisit soigneusement les ports vers lesquels une trame est recopiée. Lorsqu'une trame arrive sur un *switch* depuis un port p, le *switch* la recopie sur tous ses autres ports sauf le port p ; cependant, il en profite pour stocker l'information que l'adresse A de l'émetteur se trouve sur le port p. Lorsque, plus tard, une trame de B destinée à A arrive sur un port q, le *switch* ne recopie la trame que vers le port p, ou se trouve le destinataire ; il en profite aussi pour retenir l'information que B se trouve sur le port q. Ainsi, après la première trame, les trames *unicast* ne sont plus recopiées.

Micro-segmentation Initialement, les *switches* étaient chers et les réseaux Ethernet étaient constitués principalement de *hubs*, avec juste quelques *switches* placés aux endroits stratégiques du réseau. Depuis, le coût des *switches* a baissé à un tel point qu'on ne trouve plus de *hubs*, et les Ethernet modernes sont entièrement constitués de *switches*. Dans un tel réseau, tous les liens sont point-à-point, et le mécanisme CSMCA/CD n'est plus utile : les liens utilisent un protocole *full-duplex* et les collisions ne sont plus possibles. On dit qu'un tel réseau est *micro-segmenté*.

Protocole de l'arbre couvrant A priori, un réseau *switché* doit utiliser une topologie acyclique. Les bons *switches* implémentent le protocole de l'arbre couvrant *(STP, Spanned Tree Protocol)* qui rend certains ports inactifs pour construire une topologie acyclique (un arbre couvrant du graphe d'adjacence des *switches*).

STP est très simple : ce qu'il construit est un arbre, ce qui rend le traffic non-optimal, et cause de la congestion à certains endroits du réseau (notamment à la racine). Il n'est donc toujours pas une bonne idée de construire de très gros liens Ethernet : au delà de quelques centaines de nœuds, il vaut mieux scinder le réseau en plusieurs liens interconnectés à la couche 3 par des *routeurs* qui, eux, sont conçus pour pouvoir passer à l'échelle.