# Protocoles Réseau IV

Juliusz Chroboczek

4 octobre 2022

La couche réseau de TCP/IP est une *couche de convergence* : il n'y a en principe qu'un seul protocole de couche réseau qui transporte des données utilisateur, et ce même protocole est employé quel que soit le protocole de couche lien utilisé.

La couche réseau fournit un service de communication de bout en bout par paquets, non-fiable, non-ordonné. Ce service a donc les mêmes caractéristiques que celui qui est fourni par la couche lien, mais il est de bout en bout au lieu d'être local au lien. La principale difficulté à cette couche est le passage à l'échelle : alors que la couche lien fournit un service à 10, 100, au plus 1000 nœuds, la couche réseau doit être capable de desservir des internets de plusieurs milliards de nœuds.

Contrairement au principe énoncé ci-dessus, la couche réseau contient en ce moment deux protocoles : IP classique (IPv4) et IPv6. Ces deux protocoles fournissent des services semblables, et les mêmes protocoles de couche transport, avec des modifications mineures, fonctionnent aussi bien au-dessus de IPv4 que IPv6.

## 1 Routage *next-hop*

Il existe plusieurs paradigmes de routage [1].

**Routage par la source** En *routage par la source (source routing)*, l'émetteur calcule la route complète vers le destinataire, et la stocke dans l'entête de paquet. L'entête contient aussi un pointeur qui indique le routeur qui traite en ce moment un paquet. Lorsqu'il fait suivre un paquet, un routeur avance simplement le pointeur qui indique alors le voisin auquel il faut faire suivre le paquet.

Le routage par la source a beaucoup d'avantages : il est simple d'un point de vue théorique (par exemple, il est trivial de montrer qu'un paquet routé par la source ne boucle pas indéfiniment), il ne demande pas d'état dans les routeurs, et l'opération de *forwarding* est très simple et donc efficace. Cependant, il a deux défauts majeurs : il demande des entêtes de grande taille, et il donne trop de contrôle à l'émetteur (qui peut par exemple ignorer les politiques commerciales des fournisseurs). Pour ces raisons, il est peu utilisé, au moins à la couche réseau.

1. Le terme français *routage* est la traduction de deux termes anglais distincts : *forwarding* et *routing*. Dans ce paragraphe, il s'agit de *forwarding*, le *routing* sera traité au cours suivant.

**Routage par commutation de circuits** La *commutation de circuits* [2] est une version plus économique du routage par la source. En commutation de circuits, l'émetteur calcule la route complète, puis lui affecte un identificateur unique, le *numéro de circuit*. Il communique ensuite avec chacun des routeurs sur la route pour lui indiquer le voisin auquel faire suivre les paquets identifiés par ce numéro de circuit. L'entête ne contient que le numéro de circuit.

L'avantage principal de la commutation de circuits est qu'elle permet d'avoir un entête minuscule, ce qui la rend particulièrement adaptée au traffic de voix. Cependant, la commutation de circuits introduit de l'état au sein du réseau, ce qui la rend fragile et peu résiliente aux pannes. Elle était utilisée dans les réseaux de télécommunications jusqu'à récemment (LTE ne s'en sert presque plus).

**Routage *next-hop*** En *routage next-hop*, chaque nœud calcule une *table de routage* qui associe à chaque destination un *next hop*, l'adresse du voisin auquel envoyer les paquets. Comme la table de routage couvre des milliards de destinations, elle est stockée de manière structurée, chacune de ses entrées couvre tout un *préfixe* d'adresses.

Le routage *next-hop* est un paradigme simple et efficace, c'est lui qui est utilisé sur l'Internet. Ses principales limitations sont qu'il est difficile à rendre correct (les routes sont construites par concaténation des *next hops*, et il n'est pas évident qu'elles ne contiennent pas de cycles), et qu'il est peu flexible.

# 2 Adressage

Chaque *interface* (carte réseau) est identifiée par une *adresse* qui est en principe globalement unique (mais voyez la description du NAT ci-dessous).

## 2.1 IPv4

IPv4 adresse les interfaces par des *adresses IP*, d'une longueur de 32 bits (4 octets), qui sont en principe globalement uniques. On note les adresses IP comme quatre nombres en base 10 séparés par des points, par exemple `134.157.168.57`.

**Adresses spéciales** Certaines adresses IP sont spéciales :

- `0.0.0.0` est l'adresse *indéfinie*. Elle n'apparaît normalement jamais sur le fil [3], et sert dans les API à représenter soit « n'importe quelle adresse », soit « pas d'adresse » ;
- `127.0.0.1` est l'adresse *loopback*. Elle sert à représenter la notion de « cette machine-ci » ;
- `255.255.255.255` est l'adresse *limited broadcast*. Elle sert comme adresse de destination d'un paquet destiné à toutes les machines du lien local [4].

2. À ne pas confondre avec le paradigme de commutatin décrit au premier cours qui porte exactement le même nom. Ici, il s'agit d'un paradigme de routage dans les réseaux à commutation de paquets.

3. Le protocole DHCPv4 viole cette règle.

4. Lequel ? IPv4 n'est pas conçu pour supporter les hôtes *multihomed*, connectés à plusieurs liens simultanément. On peut contourner cette limitation, mais il vaut mieux passer à IPv6.

### 2.2 Adressage IPv6

Les adresses IPv6 ont une longueur de 128 bits, et elles sont notées comme des suites de groupes de 16 bits en hexadécimal, séparées par des « `:` ». Par exemple, ma machine de bureau avait jadis l'adresse IPv6 suivante :

```
2001:660:3301:8061:21c:25ff:fecf:7973
```

Une suite de zéros peut être omise à l'intérieur d'une adresse. Par exemple, les deux notations suivantes représentent la même adresse :

```
2001:660:3301:8063:0:0:0:1
2001:660:3301:8063::1
```

**Adresses spéciales** IPv6 définit un certain nombre d'adresses spéciales :

- `::` (128 bits de zéros), est l'adresse indéfinie ;
- `::1` (127 bits de zéros, un bit valant 1) est l'adresse *loopback*.

Il n'y a pas en IPv6 d'adresses de *broadcast* : le mécanisme de multidiffusion est différent.

## 3 Préfixes et sous-réseaux

**Préfixes** Un *préfixe* est un ensemble d'adresses dont la taille est une puissance de deux et dont la première adresse est un multiple de la taille. Par exemple, l'ensemble

$$\{134.157.168.0, 134.157.168.1, \ldots 134.157.168.127\}$$

est un préfixe contenant $2^7$ adresses.

Les adresses contenues dans un préfixe commencent par un certain nombre de bits constants suivis d'un nombre de bits qui varient. On appelle *longueur* du préfixe le nombre de bits qui ne varient pas — par exemple, le préfixe ci-dessus a une longueur de 25 bits. (Un préfixe long est donc petit, et un préfixe court est gros.)

On note un préfixe par la première adresse du préfixe suivie d'un *slash* « `/` » suivi de la longueur du préfixe. Le préfixe ci-dessus s'écrit `134.157.168.0/25`.

Une notation obsolète consiste à représenter un préfixe par une paire *adresse et masque*, où le masque est un entier de 32 bits où les bits du préfixe sont indiqués par des 1 ; par exemple, un `/24` correspond à un masque valant `255.255.255.0` ; un `/25` à `255.255.255.128`, etc. Enfin, on appelait jadis *réseau de classe A* un `/8`, *classe B* un `/16` et *classe C* un `/24`. Cette terminologie est aujourd'hui obsolète, mais vous entendrez parfois les vieux barbus l'employer.

**Sous-réseaux** Un *sous-réseau (subnet)* ou simplement *réseau* est l'ensemble des nœuds connectés à un lien. Dans l'architecture Internet, on affecte à un sous-réseau des adresses contenues dans un ou plusieurs préfixes disjoints des préfixes des autres sous-réseaux. Cette structure améliore le passage à l'échelle du plan de contrôle.

# 4 Format des paquets

## 4.1 Format des paquets IPv4

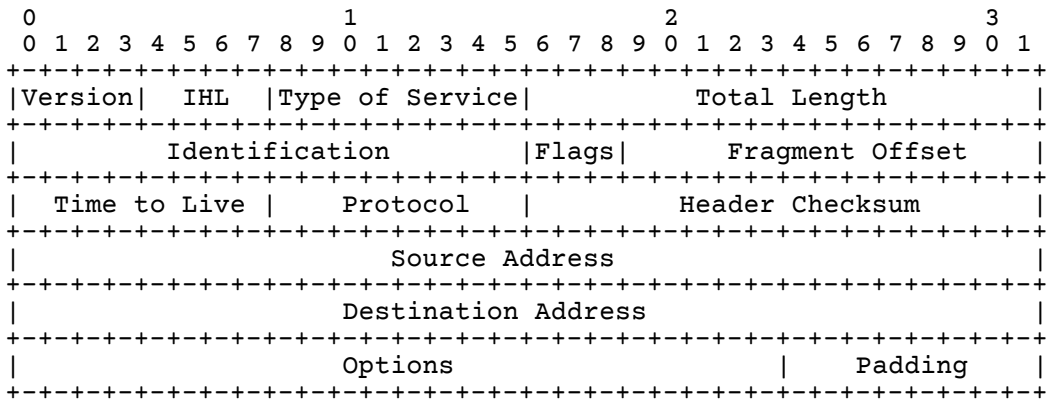


FIGURE 1 — Format de l'entête IPv4

Un paquet IPv4 consiste d'un entête (figure 1) suivi des données de couche supérieure. L'entête IPv4 contient en particulier :

- les adresses IP source et destination (*Source Address*, *Destination Address*) ;
- le numéro du protocole de couche supérieure contenu dans le paquet IP *(Protocol)* ;
- le nombre maximal de sauts *(Time to Live)* que le paquet peut parcourir, ce qui permet de briser les boucles de routage ;
- la longueur totale du paquet *(Total Length)*, qui permet de déterminer la longueur des données de couche supérieure sans dépendre de la couche lien.

## 4.2 Format des paquets IPv6

La structure du paquet IPv6 (figure 2) est plus simple que celle d'un paquet IPv4. La principale différence est que les adresses source et destination passent à 128 bits, que certains champs ont été omis (il n'y a plus de *checksum*, on fait confiance aux couches basses pour ne pas corrompre l'entête), et d'autres renommés (le champ *TTL* s'appelle maintenant *Hop Limit*, ce qui est plus explicite mais moins joli). Un champ a été ajouté : le *Flow Label*, qui permet aux routeurs d'identifier les flots sans consulter les données de couche transport ou application.

# 5 Opération du protocole

**Vue d'ensemble** Une application sur le nœud **A** désire envoyer des données à une application sur le nœud **Z** (figure 3). Cependant, **A** et **Z** sont séparés par des routeurs intermédiaires, **B**, **C** etc.

L'application fait un appel à `write` ou à `sendto`, ce qui a pour effet de remettre les données à la couche transport. La couche transport fait sa cuisine (par exemple TCP va *segmenter* les données),

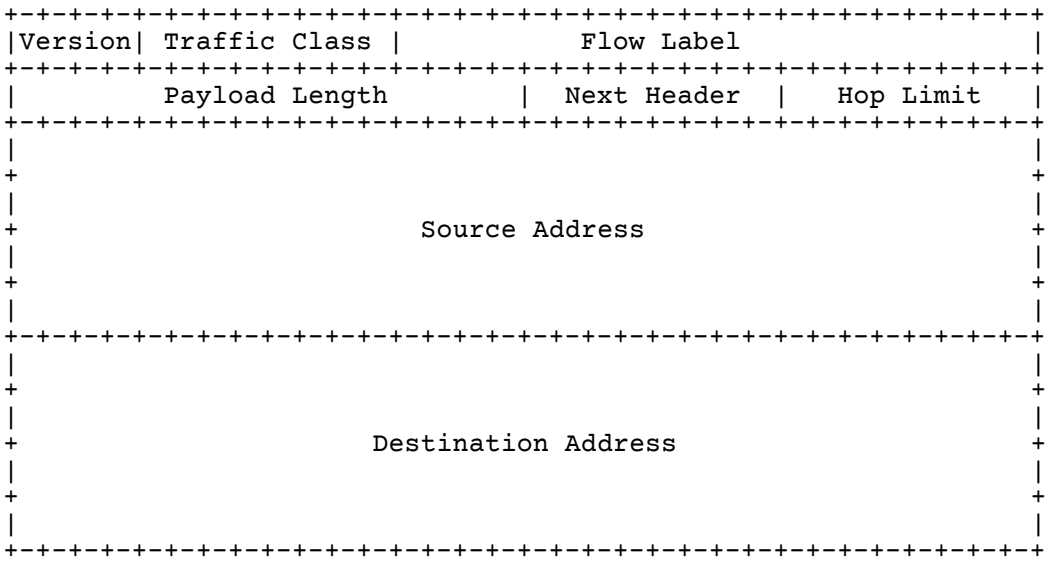


FIGURE 2 — Format de l'entête IPv6

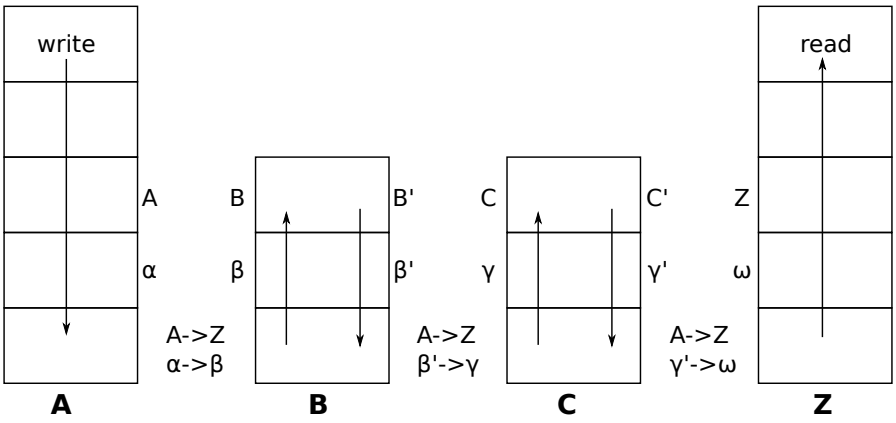


FIGURE 3 — Transfert de données

puis remet les données à la couche réseau. La couche réseau ajoute un entête IP avec source égale à $A$ (l'adresse IP d'une des interfaces de **A**) et destination égale à $Z$ (l'adresse IP d'une des interfaces de **Z**). La couche réseau consulte sa table de routage, détermine que le paquet doit être transféré à l'interface d'adresse $B$ sur **B** ; elle détermine son adresse de couche lien (adresse MAC) $\beta$, puis remet le paquet à la couche lien.

La couche lien de **A** transmet une trame contenant le paquet à destination de $\beta$ ; à la couche lien, cette trame a l'adresse source $\alpha$ et la destination $\beta$, mais elle contient un paquet de source $A$ et destination $Z$.

Lorsque la trame est remise à **B**, le paquet est décapsulé et remis à la couche réseau ; la couche réseau détermine que le paquet doit être transféré à l'interface d'adresse $C$ sur **C**, et le processus se répète ; cette fois-ci, la trame aura pour source $\beta'$ et pour destination $\gamma$, mais elle contiendra encore un paquet de source $A$ et de destination $Z$.

Deux choses à retenir :

- les adresses de couche réseau sont de bout en bout (elles ne changent pas lors du transfert), tandis que les adresses de couche lien sont locales au lien (elles changent à chaque saut) ;
- dans un routeur intermédiaire, le paquet ne monte que jusqu'à la couche réseau ; conceptuellement, les routeurs ne contiennent que trois couches.

**Réception d'un paquet** Lorsque la couche lien indique qu'un paquet est arrivé, la couche IP commence par vérifier que le champ *checksum* est correct et que le champ *TTL* est supérieur ou égal à 1. Si ce n'est pas le cas, le paquet est ignoré *(dropped)*. Sinon, la couche IP détermine si le paquet est destiné au nœud local (l'adresse IP de destination est une des adresses du nœud local) ; si c'est le cas, le champ *Protocol* est consulté, et le paquet est remis au bon protocole de couche transport.

Si le paquet n'est pas destiné au nœud local, et celui-ci est configuré comme routeur, le champ *TTL* est décrémenté ; s'il est encore supérieur ou égal à 1, le paquet est réémis.

**Émission d'un paquet** Lorsqu'un paquet est émis, soit parce qu'il a été localement remis à la couche réseau par un protocole de couche supérieure, soit parce le nœud est un routeur et que le paquet est réémis, le protocole IP commence par consulter sa *tables de routage* (qui seront décrites plus tard) pour déterminer le *next hop* auquel envoyer le paquet — une paire (interface, IP).

La couche IP détermine ensuite l'adresse de couche lien du *next hop* (voir paragraphe 6 ci-dessous), calcule le *checksum* du paquet, et le passe à la couche lien.

## 6 Découverte de voisins

Le protocole de découverte de voisins d'IPv4, *Adress Resolution Protocol* (ARP), sert à déterminer l'adresse de couche lien d'une interface dont on connait l'adresse IP. ARP consiste d'une requête (« who-has ») et une réponse (« i-have »).

Lorsqu'un noœd A a besoin de déterminer l'adresse lien d'une interface B sur le lien local, il envoie une requête « who-has » à l'adresse lien de multidifusion. Si B reçoit cette requête, il répond avec une réponse « i-have » qui contient son adresse de couche lien. Comme le protocole

de couche lien n'est normalement pas fiable, la requête doit être répétée un petit nombre de fois si personne ne répond.

Tous les nœuds maintiennent un « cache ARP » qui contient les adresses de couche lien connues. Une entrée de ce cache est normalement purgée au bout de quelques minutes, ce qui devrait en principe forcer un nouvel échange ARP. Cependant, les protocoles de couche transport « rafraîchissent » une entrées du cache lorsqu'ils reçoivent une preuve de la correction de celle-ci (typiquement un acquittement). Il s'agit là d'un cas d'interaction entre couches particulièrement propre : l'interface entre couches est bien définie, il n'y a pas de concepts qui apparaissent à la mauvaise couche, et l'interaction n'est qu'une optimisation.

En IPv6, le protocole ARP a été remplacé par un protocole similaire, nommé *Neighbour Discovery* (ND), mais dont les messages sont transportés par ICMPv6 au lieu d'un protocole de couche lien.

## 7 Protocole de contrôle

*Internet Control Message Protocol* (ICMP) est un protocole qui sert à transmettre des indications d'erreur entre les nœuds IP, ce qui facilite énormément le déboggage du réseau. Par exemple, si un routeur n'a pas de route vers la destination d'un paquet, il émet un paquet ICMP vers la source du paquet qu'il n'a pu router. De même lorsqu'un hôte est destinataire d'un paquet ayant un numéro de protocole qu'il ne connaît pas.

ICMP contient aussi des messages de déboggage, par exemple les messages *Echo request* et *Echo reply* utilisés par la commande `ping`, ainsi que des paquets qui permettent aux protocoles de couche supérieure de déterminer les caractéristiques d'une route *(packet too big)*.

## 8 Traduction d'adresses

En principe, chaque adresse IPv4 est globalement unique. Or, l'espace d'adressage d'IPv4 est limité à quelques quatre milliards d'adresses, ce qui ne suffit plus aujourd'hui.

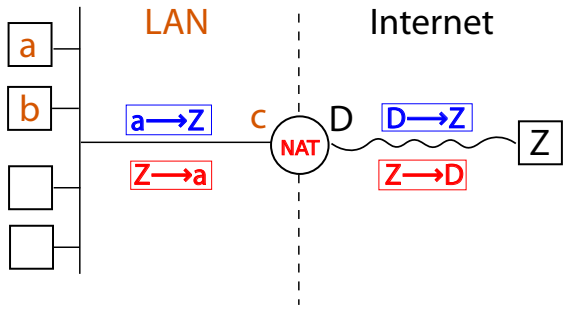


FIGURE 4 — Traduction d'adresses (NAT)

La *traduction d'adresses* (NAT, *Network Address Translation*) est une technique qui permet d'attribuer la même adresse à plusieurs machines (figure 4). À la différence d'un routeur ordinaire,

qui est symétrique, un routeur NAT divise l'Internet en un réseau local qui est « à l'intérieur » du NAT, et le reste de l'Internet, qui est « à l'extérieur ». Les paquets qui se situent à l'extérieur utilisent des adresses IP normales, dites « globales », tandis que les paquets qui sont à l'intérieur utilisent des adresses non uniques, dites « locales ». Les plages d'adresses locales officielles sont 10.0.0.0/8, 172.16.0.0/20 et 192.168.0.0/16.

On attribue une ou plusieurs adresses globales au routeur NAT. Lorsqu'un paquet transite de l'intérieur vers l'extérieur, le NAT remplace l'adresse source du paquet par l'une de ses adresses globales (il peut aussi changer le numéro de port de la couche transport) ; il retient alors l'association entre l'adresse de socket (IP, port) locale et l'adresse globale.

Lorsqu'un paquet arrive de l'extérieur, le routeur NAT recherche l'adresse de socket de destination dans sa table. Si elle est présente, il effectue la traduction inverse ; sinon, il rejette le paquet.

La technique NAT a permis de survivre à l'épuisement des adresses IPv4. Cependant, son utilisation a des conséquences graves pour l'Internet. Tout d'abord, un NAT recquiert que le premier paquet d'un flot passe de l'interieur vers l'extérieur, ce qui empêche de déployer des serveurs à l'intérieur — on revient donc à un modèle de type Minitel, où seuls les riches peuvent se permettre de déployer des serveurs. Ensuite, le NAT consulte des informations de couche transport (le numéro de port), ce qui empêche le déploiement de nouveaux protocoles. Enfin, le NAT introduit de l'état par flot à l'intérieur du réseau, ce qui rend le réseau plus fragile (moins résilient aux pannes) et empêche le routage asymétrique.

Vivement qu'on passe à IPv6.