

# Protocoles Réseau IV

Juliusz Chroboczek

27 septembre 2020

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.

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 généraux de routage. En *commutation de circuits*<sup>1</sup>, l'émetteur programme la route complète dans les routeurs puis envoie des paquets qui ne contiennent qu'un identificateur de *circuit virtuel*. En *routage par la source*, l'émetteur inclut la route complète dans les entêtes du paquet. En *routage next-hop*, chaque nœud sur la route détermine le saut suivant en fonction de sa destination, et transmet le paquet à celui-ci.

C'est le routage *next-hop* qui est utilisé sur l'Internet. La principale raison d'éviter la commutation de circuits est qu'elle stocke de l'état dans les routeurs, ce qui la rend peu résiliante aux pannes. Quant au routage par la source, il donne trop de contrôle à l'émetteur, ce qui permet à ce dernier de contourner les politiques mises en place par les opérateurs — ces derniers y sont donc opposés.

Le principal problème du routage *next-hop* est qu'il n'empêche pas les pathologies de routage, telles que les boucles et les trous noirs, et demande donc des protocoles de routage particulièrement robustes, ce que nous étudierons en détail dans le cours suivant.

---

1. À ne pas confondre avec la *commutation de circuits*, qui est un paradigme de commutation, et que nous avons vu au premier cours. Oui, les noms sont exactement les mêmes.

## 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, et ne sont jamais allouées à aucune interface :

- 0 . 0 . 0 . 0 est l'adresse *indéfinie*. Elle n'apparaît normalement jamais sur le fil<sup>2</sup>, 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 n'apparaît jamais sur le fil, et 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<sup>3</sup>.

### 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:feef: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.

---

2. Le protocole DHCPv4 viole cette règle.

3. Lequel? IPv4 n'est pas conçu pour supporter les hôtes *multihomed*.

### 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

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

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

Un paquet du protocole 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, et d'autres renommés. (En particulier, le champ *TTL* s'appelle maintenant *Hop Limit*, ce qui est plus explicite mais moins joli.)

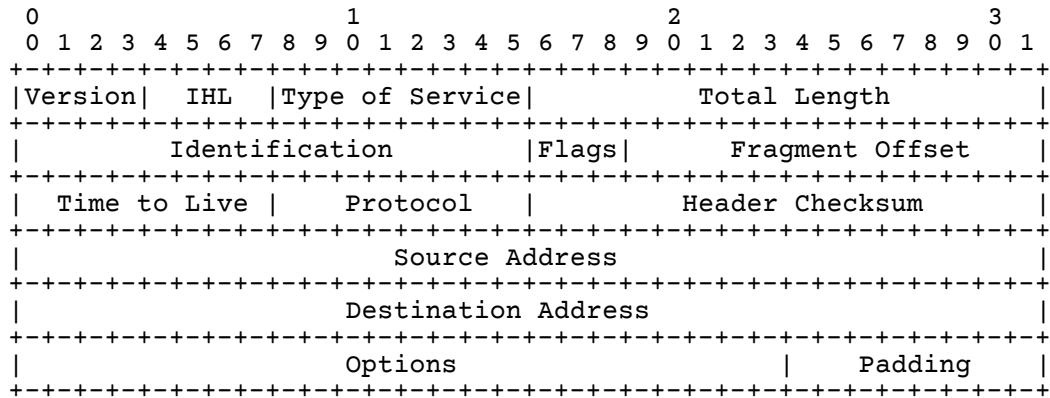


Figure 1 — Format de l'entête IPv4

## 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), 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);

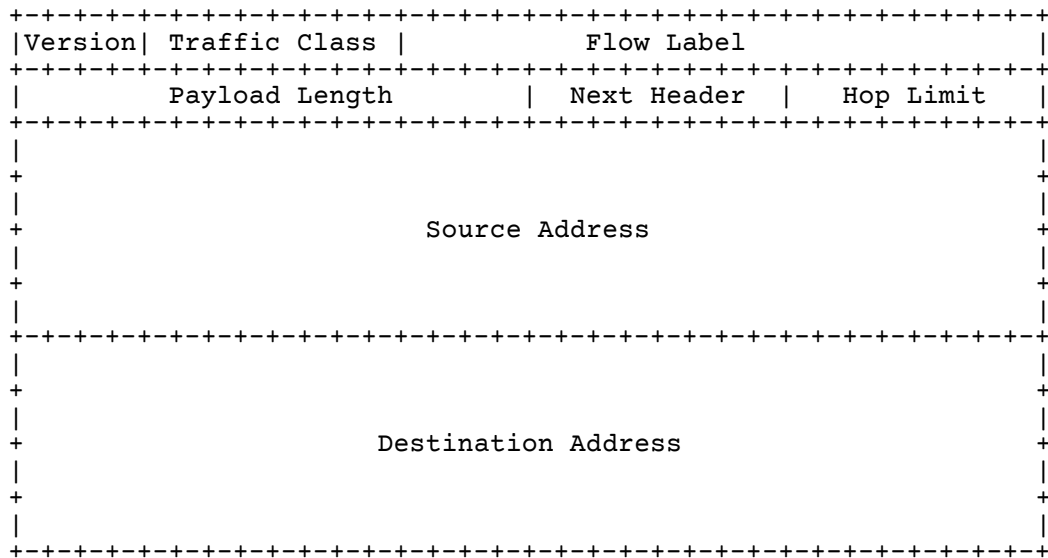


Figure 2 — Format de l'entête IPv6

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, *Address Resolution Protocol* (ARP), sert à déterminer l'adresse de couche lien d'une interface dont on connaît l'adresse IP. ARP consiste d'une requête (« who-has ») et une réponse (« i-have »).

Lorsqu'un nœud 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 multidiffusion. 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

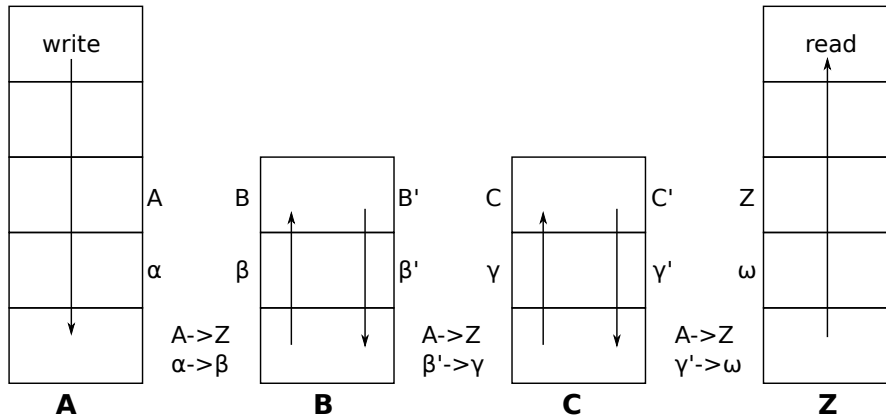


Figure 3 — Transfert de données

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 débogage 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 débogage, 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<sup>4</sup>.

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

4. Je n'en reviens pas.

métrique, 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 requiert que le premier paquet d'un flot passe de l'intérieur 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.