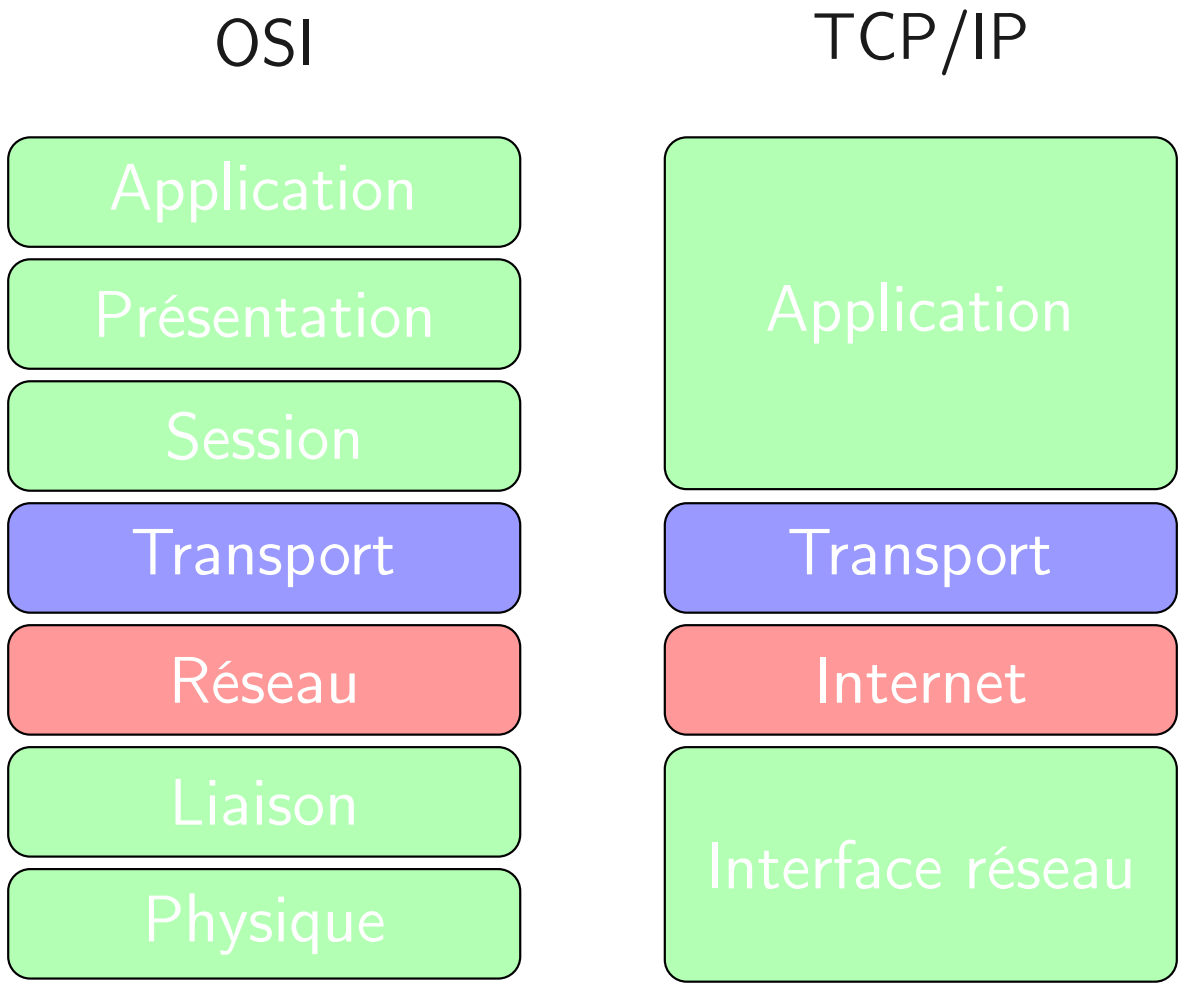
Couche Internet

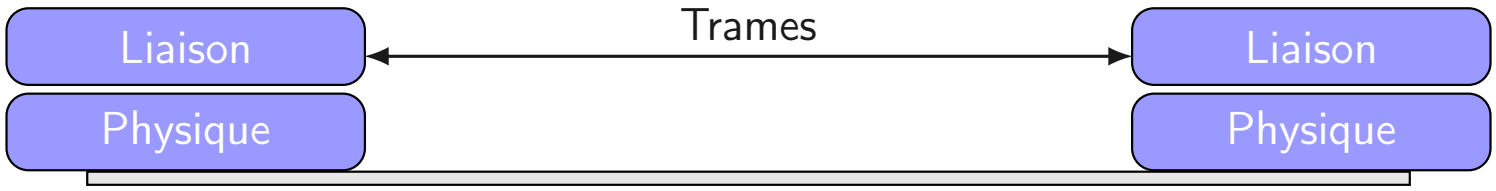
Martin Delacourt, Université d'Orléans

L3 Réseaux — 2023/2024

Les modèles

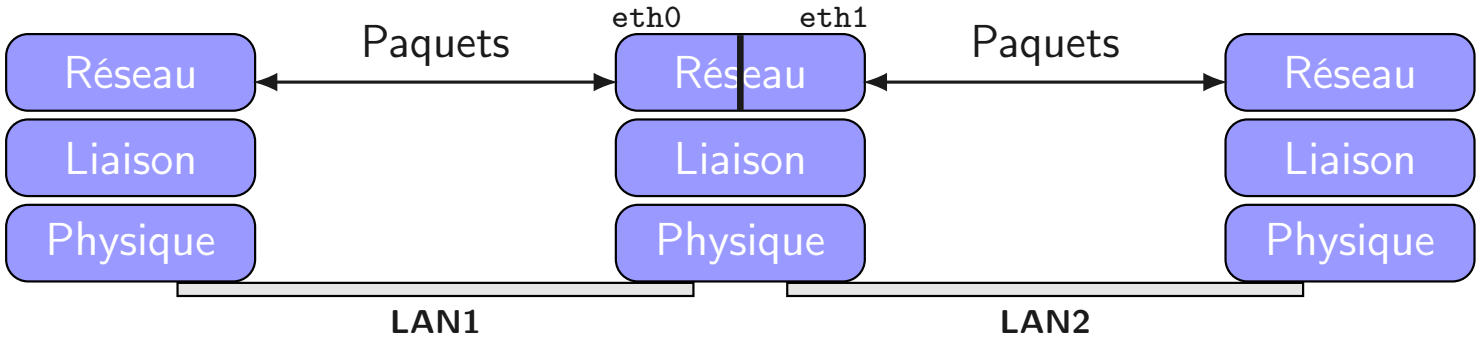

OSI
TCP/IP
Application
Présentation
Session
Transport
Réseau
Liaison
Physique
Application
Transport
Internet
Interface réseau

Couches basses : service rendu



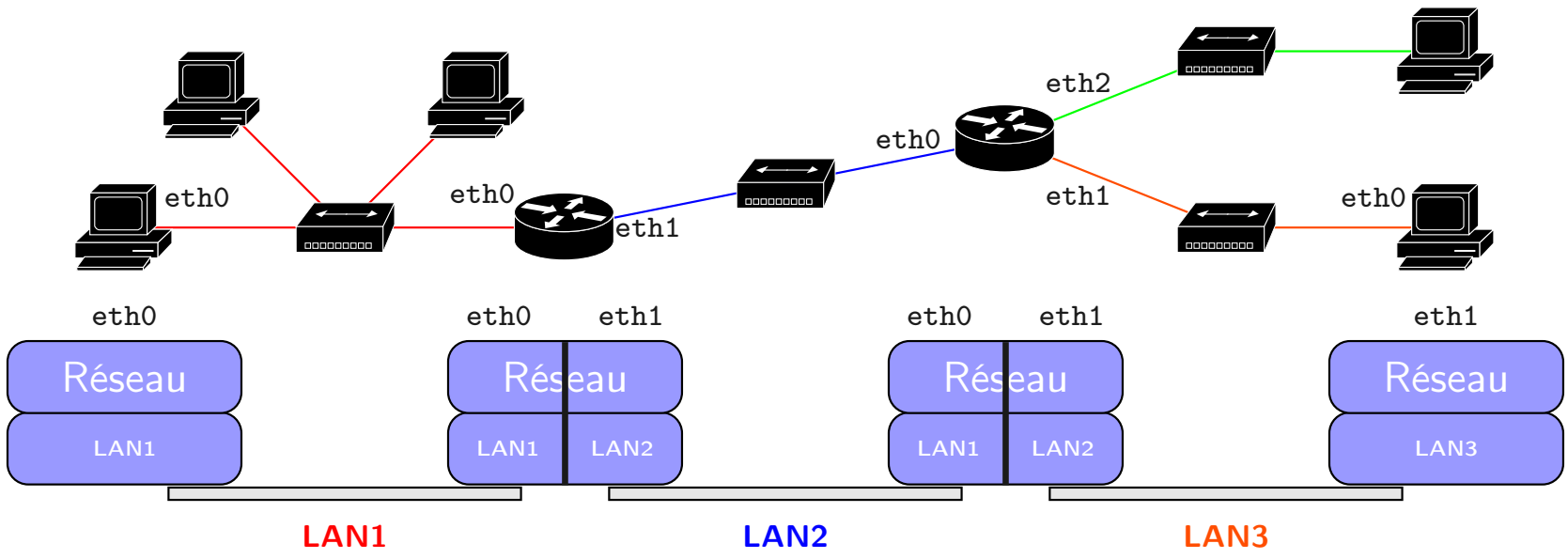
- Circulation de trames, longueur maximale fixée.
- Pas d'erreur de transmission.
- Transmission uniquement au sein d'un réseau local.

Couche réseau



- Transmission de paquets.
- Différents réseaux locaux traversés sur des médias quelconques.
- Pas de fiabilité dans le cas du protocole IP : perte, duplication, inversion.

Interconnexion de LAN



Un routeur est une machine qui possède des interfaces dans plusieurs réseaux locaux.

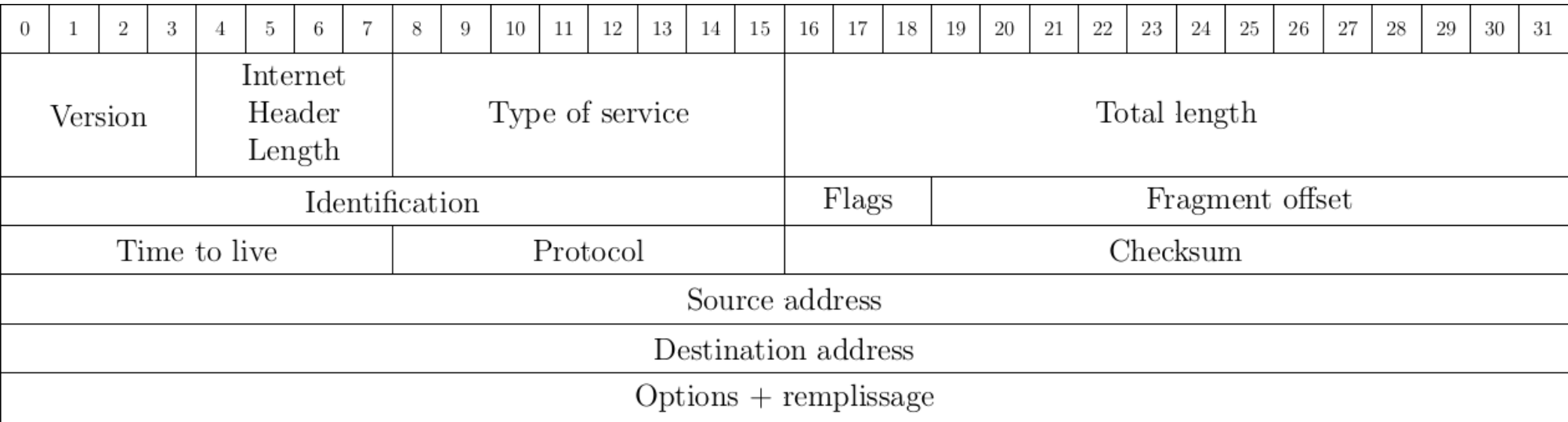
Rôle de la couche réseau

Chaque machine doit pouvoir contacter n'importe quelle autre machine sur le réseau. Pour cela, qu'elle soit sur le même réseau local ou non, il faut pouvoir :

- l'identifier : adressage IP ;
- la joindre : routage.

Au coeur d'internet, le protocole IP permet d'accomplir ces tâches. Dans ce cours : IPv4 (RFC 791), dans le TP5 : IPv6.

Adressage


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
Internet Header Length
Type of service
Total length
Identification
Flags
Fragment offset
Time to live
Protocol
Checksum
Source address
Destination address
Options + remplissage

Envoyer un paquet

Les adresses sont données aux interfaces réseaux, pas aux hôtes, une machine peut avoir plusieurs interfaces.

Connaissant une adresse, pour envoyer un paquet :

- si l'adresse est dans le réseau local, on envoie le message directement ;
- sinon, on envoie le paquet à un routeur (la passerelle) qui saura quel chemin prendre.

Il faut donc que les adresses induisent une géographie globale, et qu'elles permettent de trouver une route vers le réseau local auquel elles appartiennent.

Les adresses

Une adresse IPv4 est de la forme x.y.z.t où x, y, z et t sont des entiers entre 0 et 255, autrement dit des octets. La taille totale de l'adresse est donc 32 bits.

Exemple :

10000001		00101101		11011110		00000111
129	.	45	.	222	.	7

Une adresse IP s'interprète comme :

- un préfixe qui constitue un identifiant de réseau ;
- un suffixe qui constitue un identifiant d'hôte dans ce réseau.

Adressage CIDR

Classless Inter-Domain Routing

Un bloc CIDR est la donnée d'une adresse et d'un masque qui est une suite de k bits à 1 suivis de $32 - k$ bits à 0.

Exemple : 129.45.0.0/255.255.0.0

Une adresse appartient au réseau si le `ET bit à bit` avec le masque est égal à l'adresse du réseau :

`129.45.222.7` ?

`129.46.2.27` ?

Notation CIDR

Classless Inter-Domain Routing

Pour condenser, on note 129.45.0.0/16 pour 129.45.0.0/255.255.0.0 où $/k$ est le nombre de bits à 1 dans le masque.

Une adresse appartient donc au réseau si ses 16 premiers bits sont 129.45.

Attention au cas où le masque ne contient pas un multiple de 8 bits à 1 :

- 65.82.11.130 est dans le réseau 65.82.11.128/30 ;
- 65.82.11.130 est dans le réseau 65.64.0.0/10.

Adresses réservées

- L'`adresse du réseau` est celle qui contient tous les bits du suffixe à 0.
- L'`adresse de broadcast` est celle qui contient tous les bits du suffixe à 1.
  Exemple : 129.45.255.255

Si le masque vaut k, il y a donc $2^{32-k} - 2$ adresses disponibles dans le réseau.

L'adresse 255.255.255.255 est l'adresse de broadcast globale.

L'adresse 0.0.0.0 a diverses interprétations selon le contexte, par exemple : "toutes les adresses IP de la machine locale".

Blocs réservés

Plusieurs blocs d'adresses dont réservés à des usages particuliers, les plus notables sont :

- Le bloc 127.0.0.0/8 est un bloc pour la machine locale(`localhost`), ces adresses ne peuvent pas se voir sur internet.
- Le bloc 224.0.0.0/4 correspond au multicast.

D'autres adresses ne se voient pas sur internet, ce sont les adresses privées :

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 169.254.0.0/16

IANA

Internet Assigned Numbers Authority

C'est l'IANA qui attribue les adresses IP, la tâche est effectuée par l'ICANN (Internet Corporation for Assigned Names and Numbers) qui délègue aux RIR.

```
% dig +short www.univ-orleans.fr
194.167.30.240
% whois 194.167.30.240
...
inetnum : 194.167.30.0 - 194.167.30.255
netname : FR-ORLEANS25
...
route : 194.167.0.0/16
descr : RENATER
...
```

Attribution des adresses

A l'intérieur d'un bloc CIDR

Lorsqu'on obtient une adresse IP, il faut :

- qu'elle soit compatible avec le réseau local ;
- qu'elle soit libre.

Attribution `statique` ?

Attribution `dynamique` ?

`Autoconfiguration` ?

Attribution dynamique

Une machine demande une adresse pour un `temps limité` lorsqu'elle se connecte sur un réseau.

L'adresse `peut changer` au cours du temps.

Protocoles : BOOTP, DHCP

`DHCP` : protocole de couche application.

Autoconfiguration

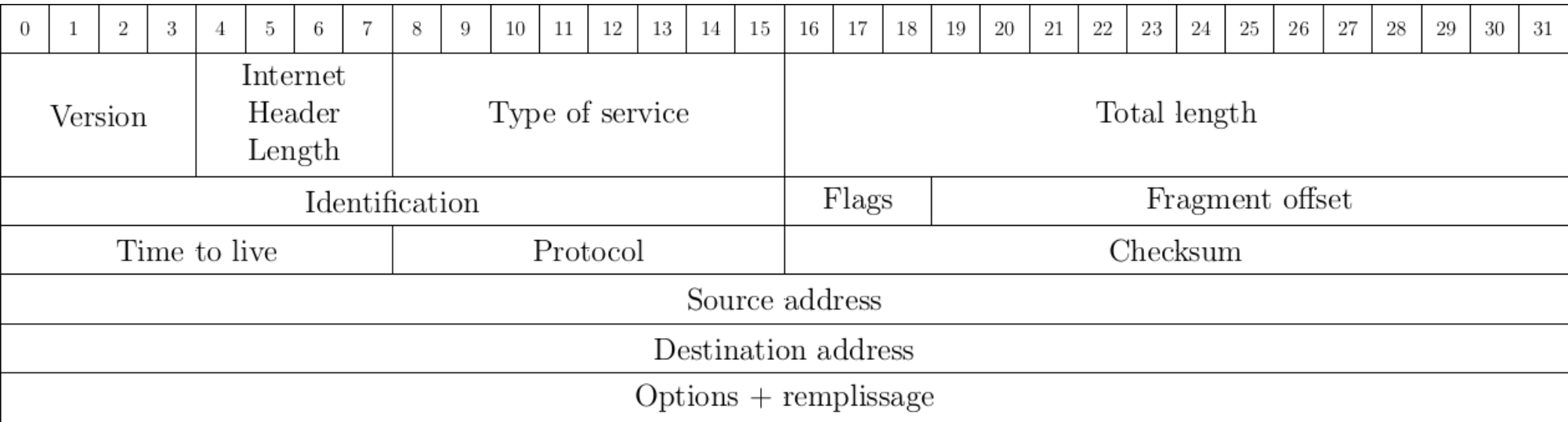
Allocation d'adresse servant uniquement dans le `réseau local`.

Bloc CIDR `169.254.0.0/16`

Chaque hôte choisit aléatoirement une adresse, et vérifie régulièrement qu'elle n'est pas utilisée (cf. Résolution)

aka `APIPA` (Automatic Private IP Addressing)

ARP


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
Internet Header Length
Type of service
Total length
Identification
Flags
Fragment offset
Time to live
Protocol
Checksum
Source address
Destination address
Options + remplissage

Réseau local

Lorsqu'on veut joindre une machine sur son réseau local :

- pas d'intermédiaire (routeur) ;
- on ne connaît que l'adresse IP ;
- il faut envoyer une trame (couche liaison) ;
- donc il faut associer une adresse MAC à l'adresse IP.

Protocole ARP

Address resolution Protocol

- Le protocole ARP obtient l'adresse MAC correspondant à une adresse IP donnée.
- On le situe entre la couche réseau et la couche liaison.
- Il peut fonctionner avec divers protocoles réseau/liaison : souvent IP/Ethernet.
- Directement encapsulé dans une trame ethernet.

Protocole ARP

Le contenu

Un paquet ARP contient les champs successifs suivants :

- HTYPE (2 octets) type de protocole liaison (1 pour Ethernet) ;
- PTYPE (2 octets) type de protocole réseau (0x800 pour IPv4) ;
- HLEN (1 octet) taille d'une adresse liaison, (6 pour Ethernet) ;
- PLEN (1 octet) taille d'une adresse réseau, (4 pour IPv4) ;
- OPER (2 octets) opération (1 = demande, 2 = réponse) ;
- SHA (h octets) adresse liaison de l'émetteur ;
- SPA (p octets) adresse réseau de l'émetteur ;
- THA (h octets) adresse liaison du destinataire ;
- TPA (p octets) adresse réseau du destinataire.

Protocole ARP

Le principe

La machine A connaît son adresse IP ipa et son adresse MAC maca, elle envoie un message à B d'adresse IP ipb, mais d'adresse MAC inconnue :

1. A envoie une demande (OPER=1) en broadcast ($ff : ff : ff : ff : ff : ff$).
2. B répond (OPER=2) uniquement à A.

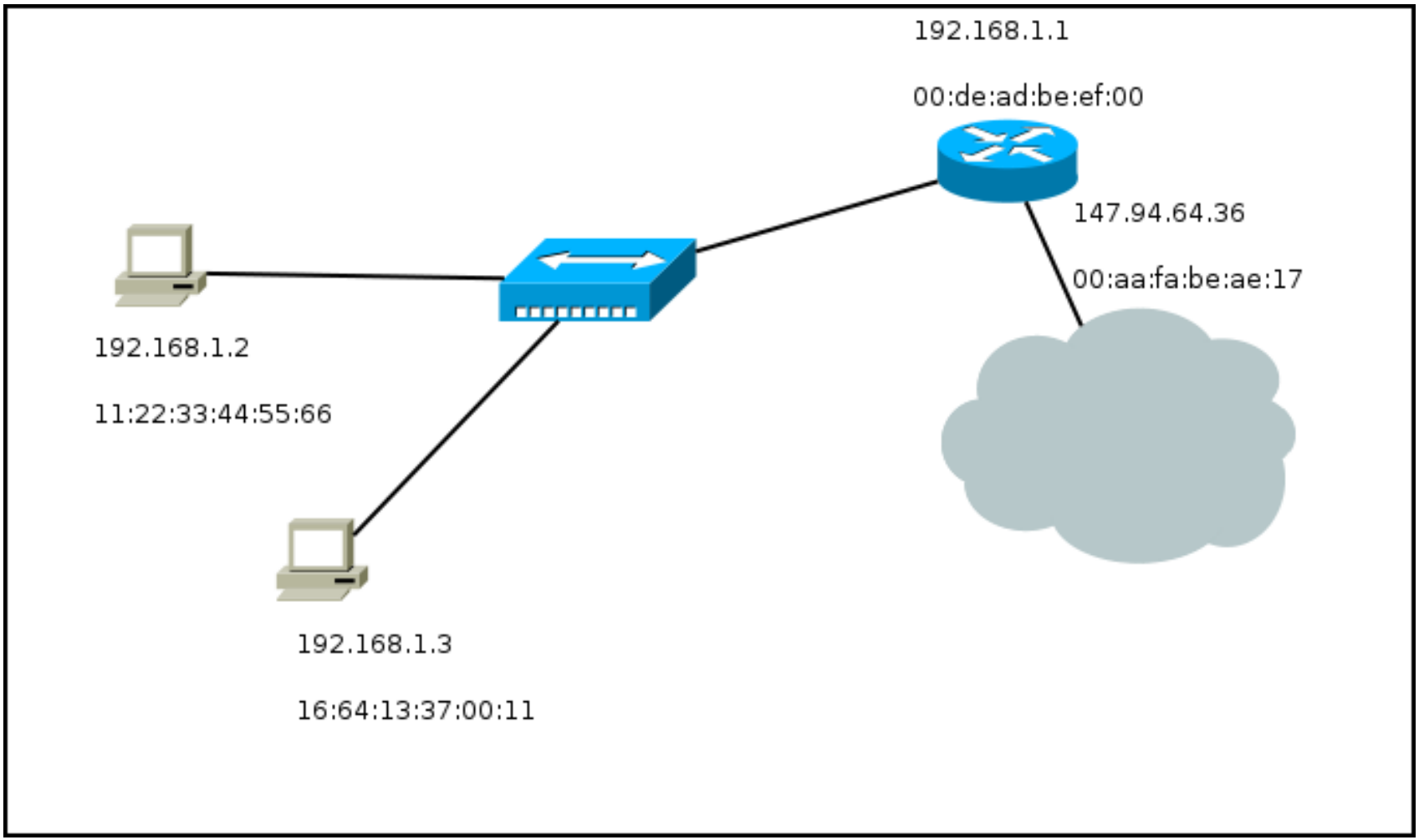
Lors de l'envoi :

- la demande contient ipa, maca (pour acheminer la réponse) et ipb, macb est à $0 : 0 : 0 : 0 : 0 : 0$;
- la réponse contient les 4 champs renseignés.

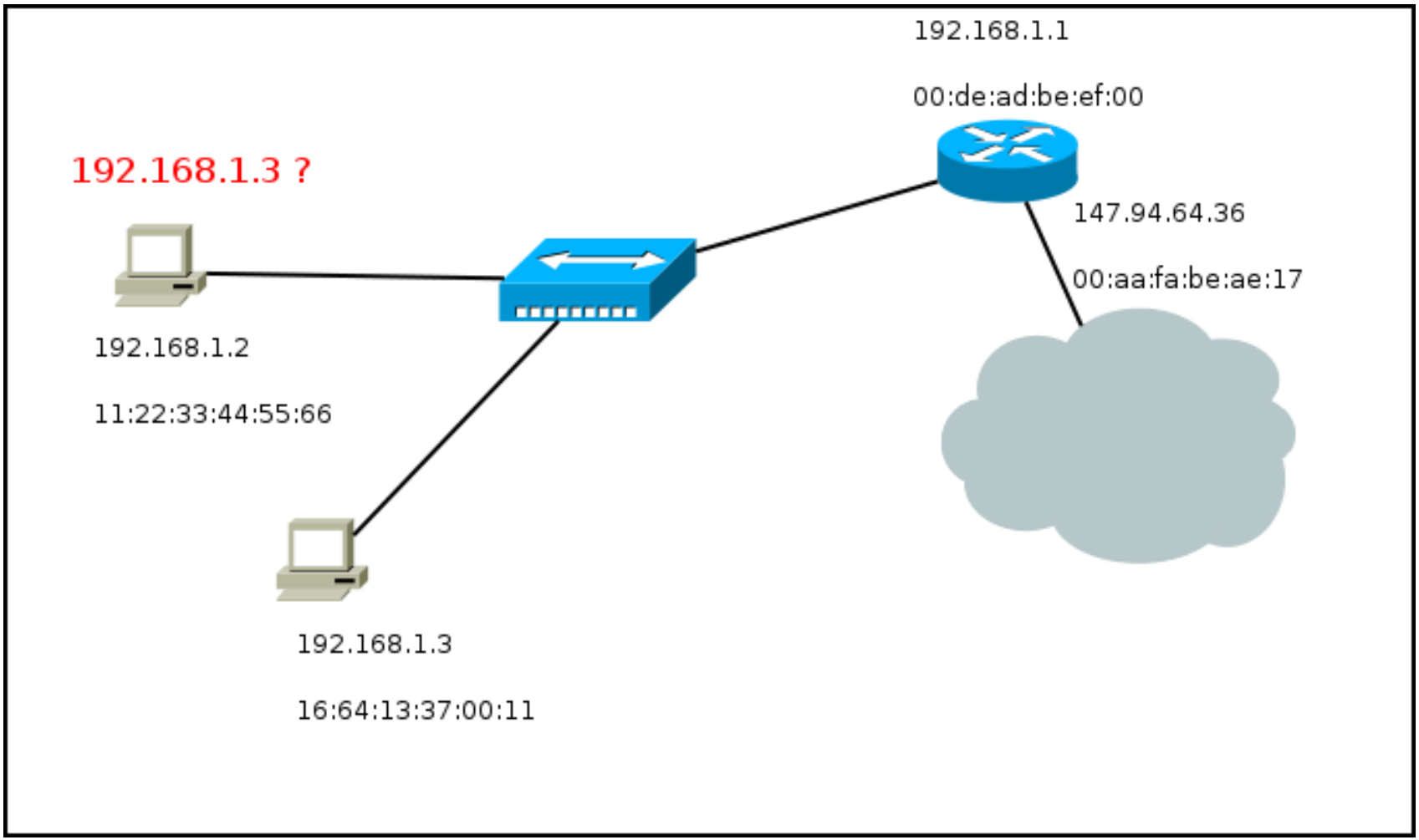
Remarques

- Système de cache avec délai d'expiration court.
- N'importe qui peut a priori répondre ! Risque d'usurpation d'identité.
- Possibilité d'envoyer le contenu de son cache sans avoir reçu de demande.
- Protocole basique d'autoconfiguration avec ipa = 0.0.0.0. pour obtenir une adresse IP.

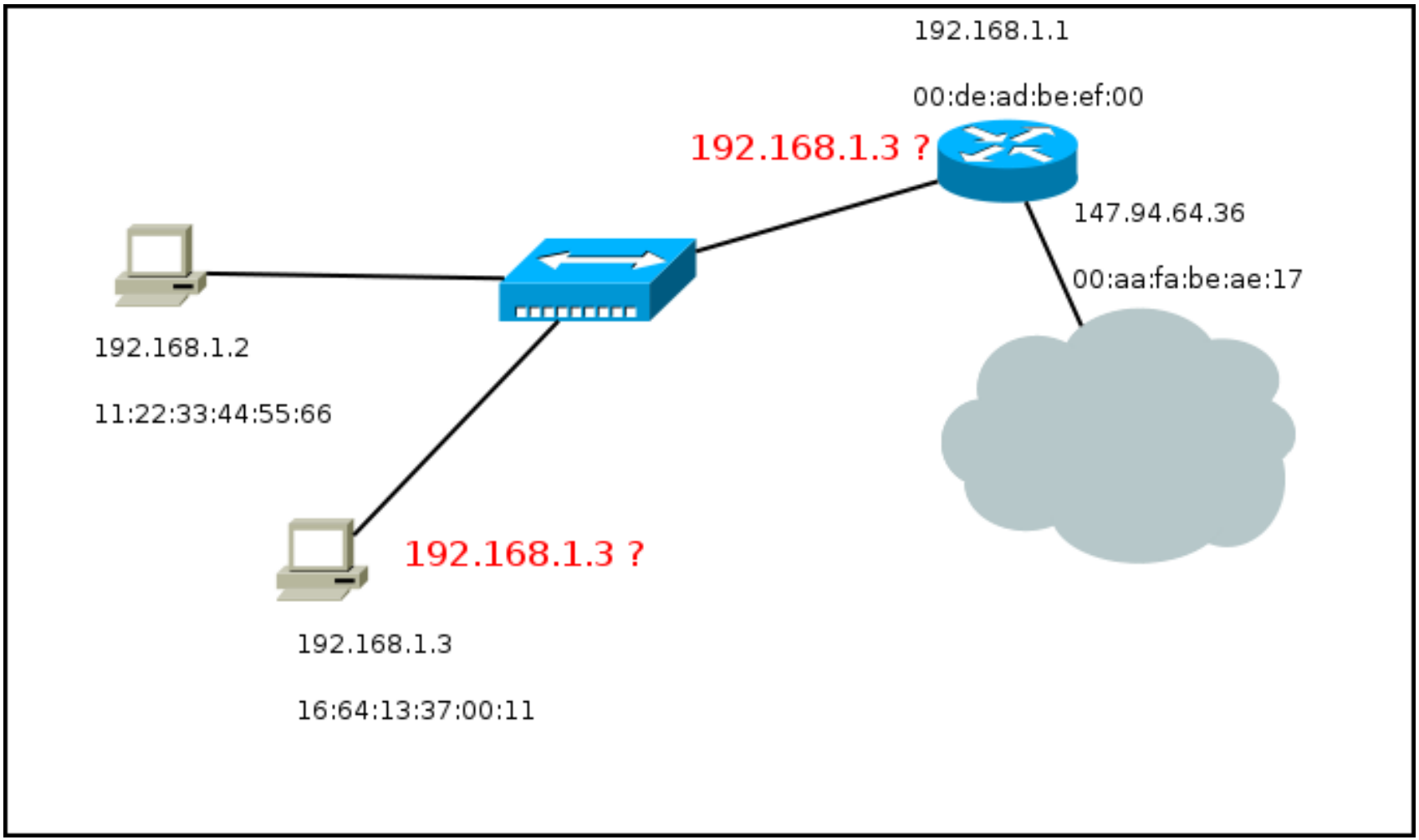
Exemple


192.168.1.1
00:de:ad:be:ef:00
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3
16:64:13:37:00:11

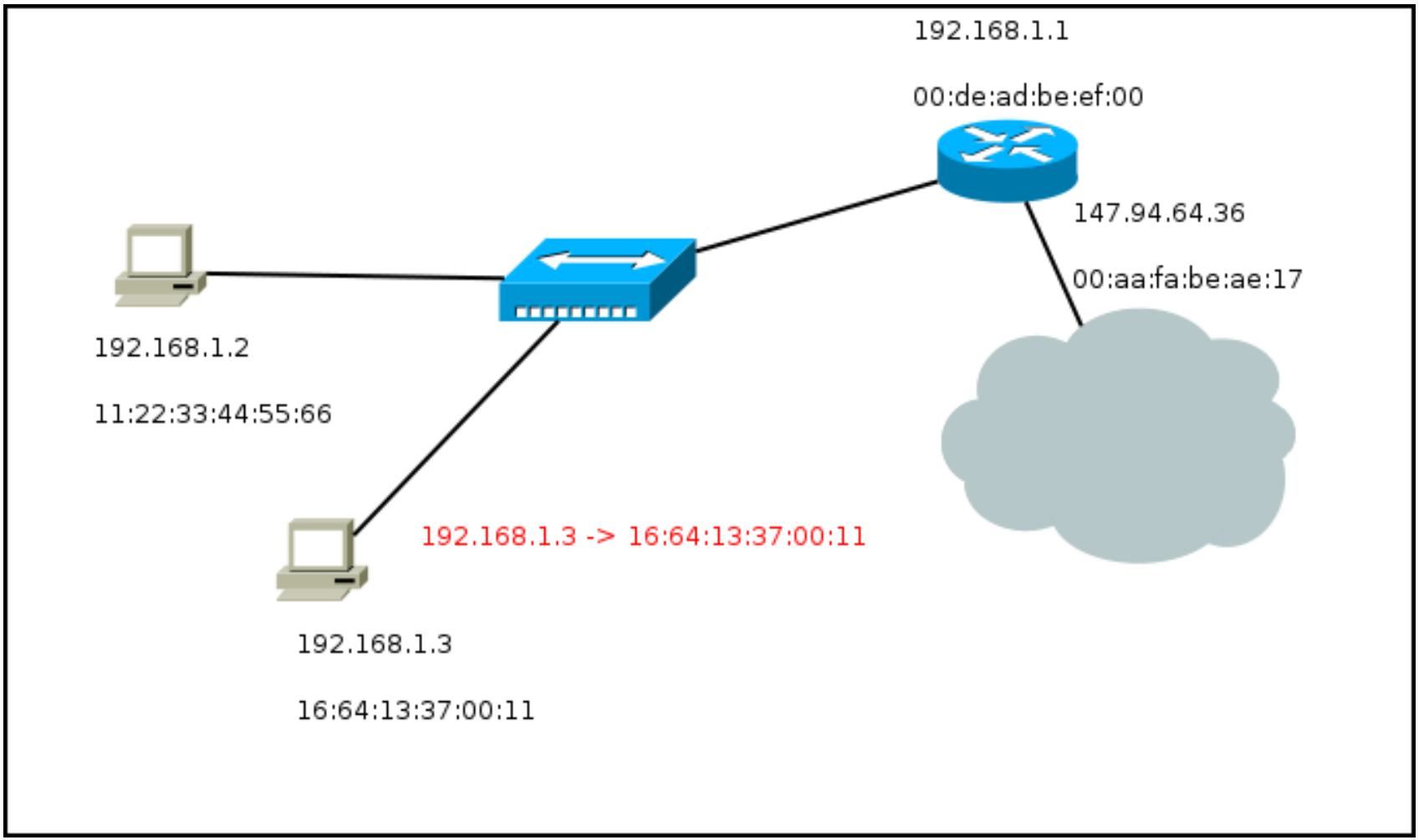
Exemple


192.168.1.1
00:de:ad:be:ef:00
192.168.1.3 ?
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3
16:64:13:37:00:11

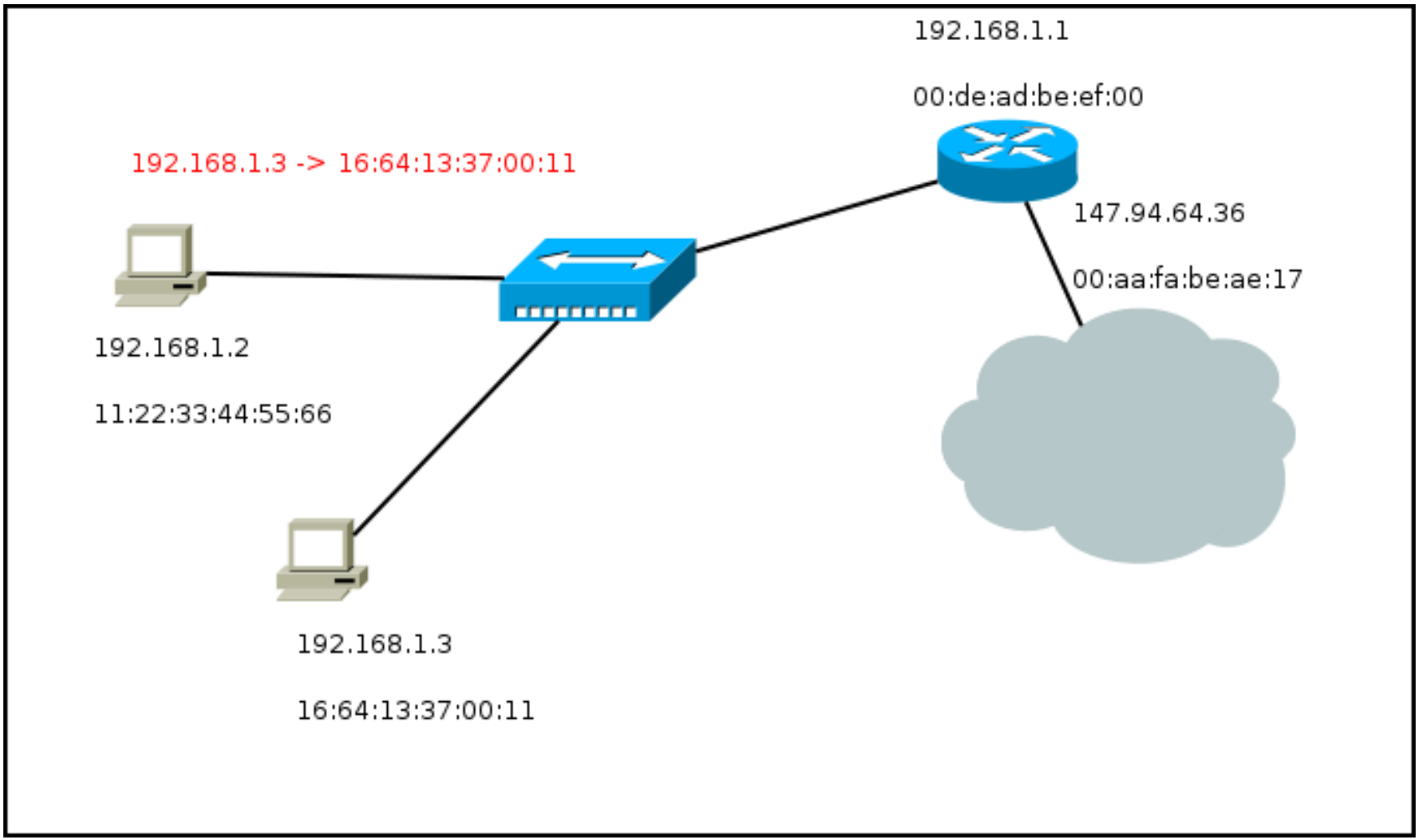
Exemple


192.168.1.1
00:de:ad:be:ef:00
192.168.1.3 ?
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3 ?
192.168.1.3
16:64:13:37:00:11

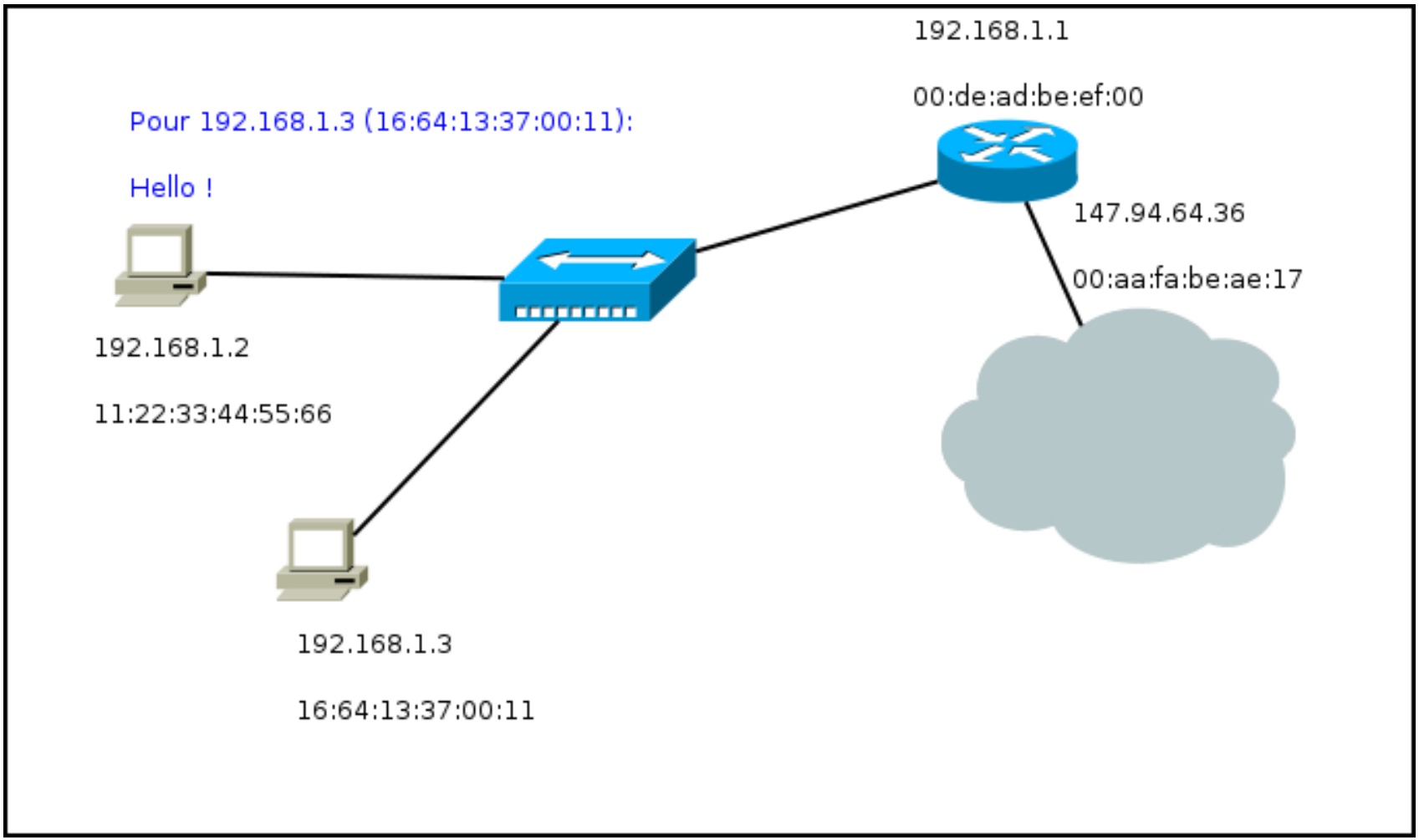
Exemple


192.168.1.1
00:de:ad:be:ef:00
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3 -> 16:64:13:37:00:11
192.168.1.3
16:64:13:37:00:11

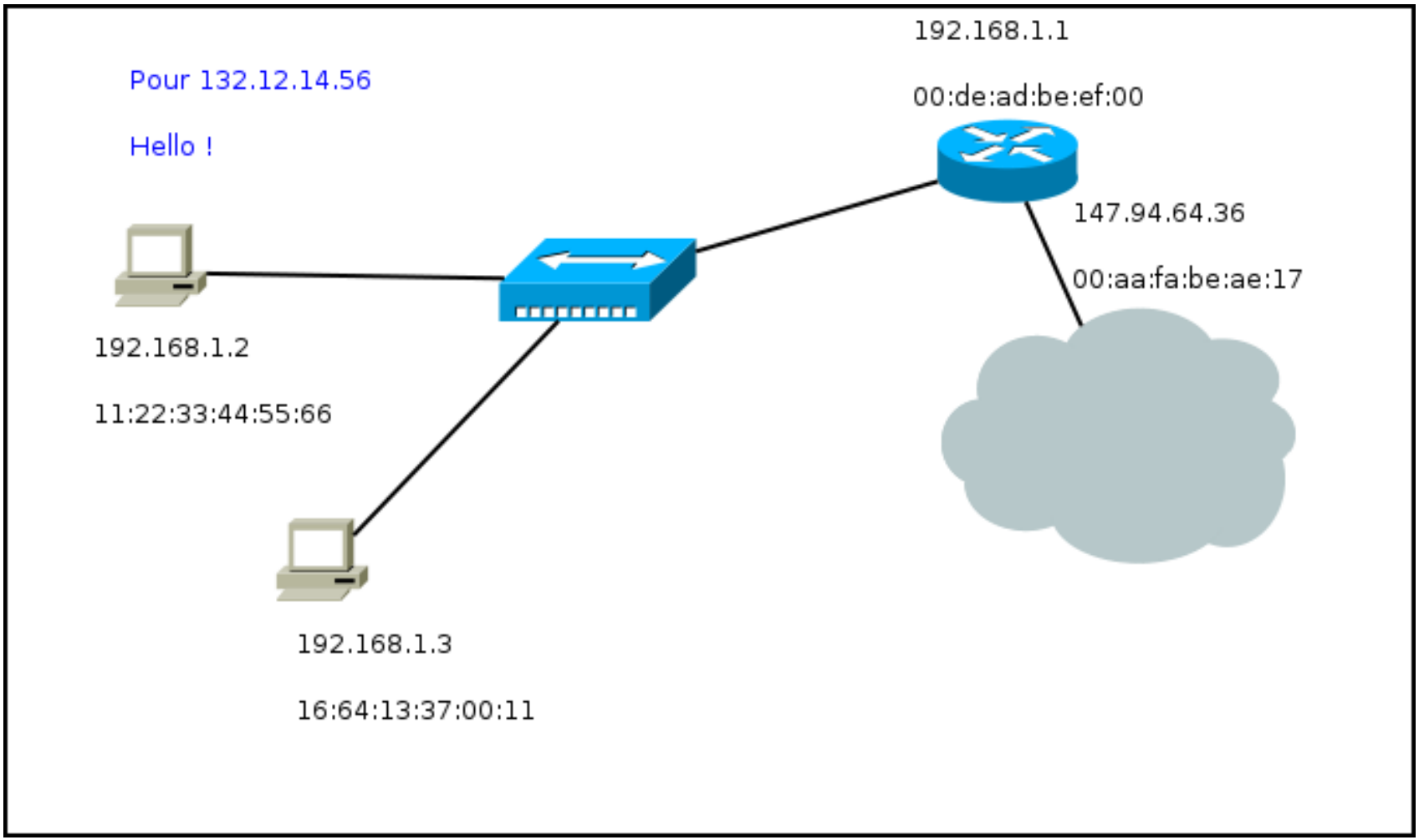
Exemple


192.168.1.1
00:de:ad:be:ef:00
192.168.1.3 -> 16:64:13:37:00:11
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3
16:64:13:37:00:11

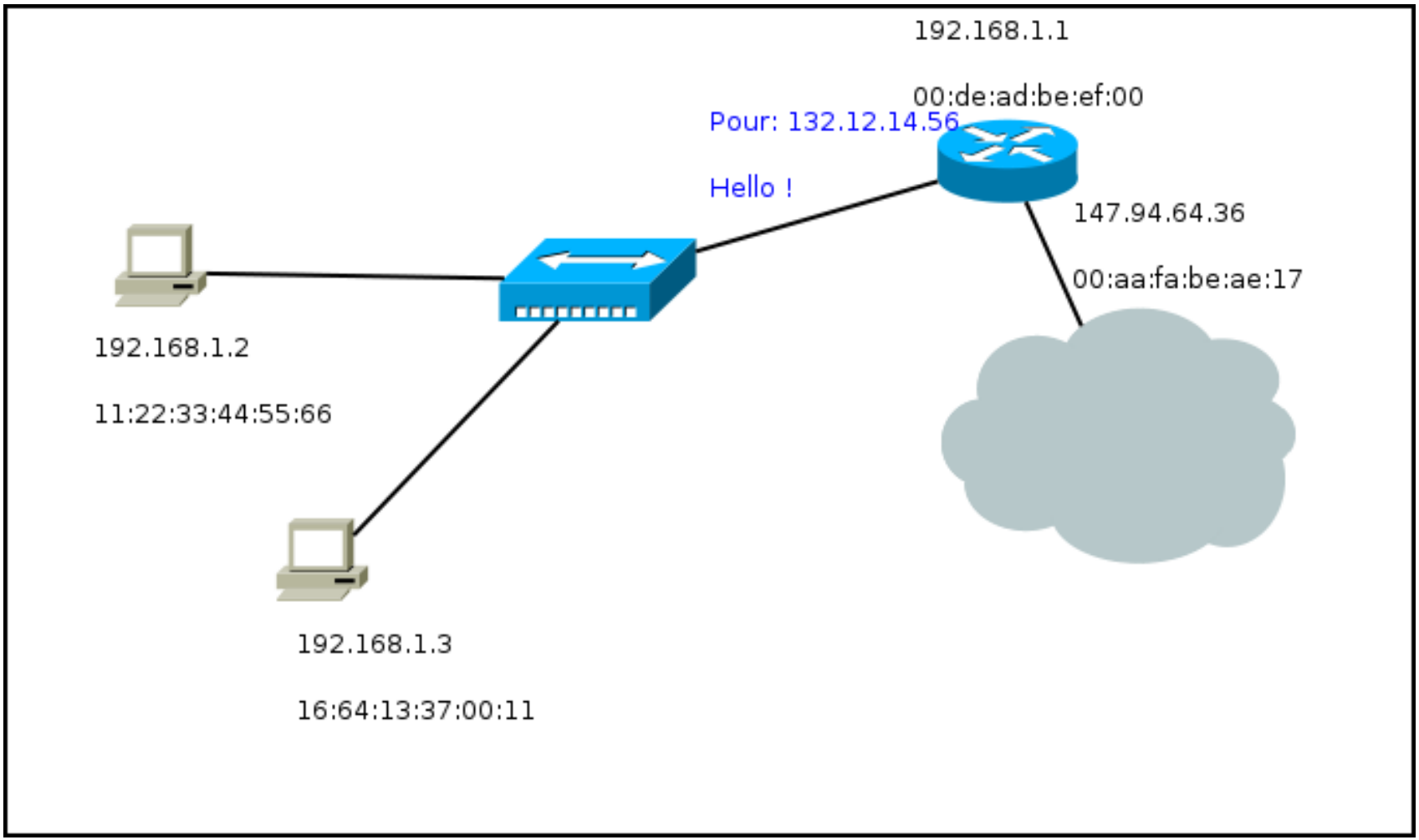
Exemple


192.168.1.1
00:de:ad:be:ef:00
Pour 192.168.1.3 (16:64:13:37:00:11):
Hello !
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3
16:64:13:37:00:11

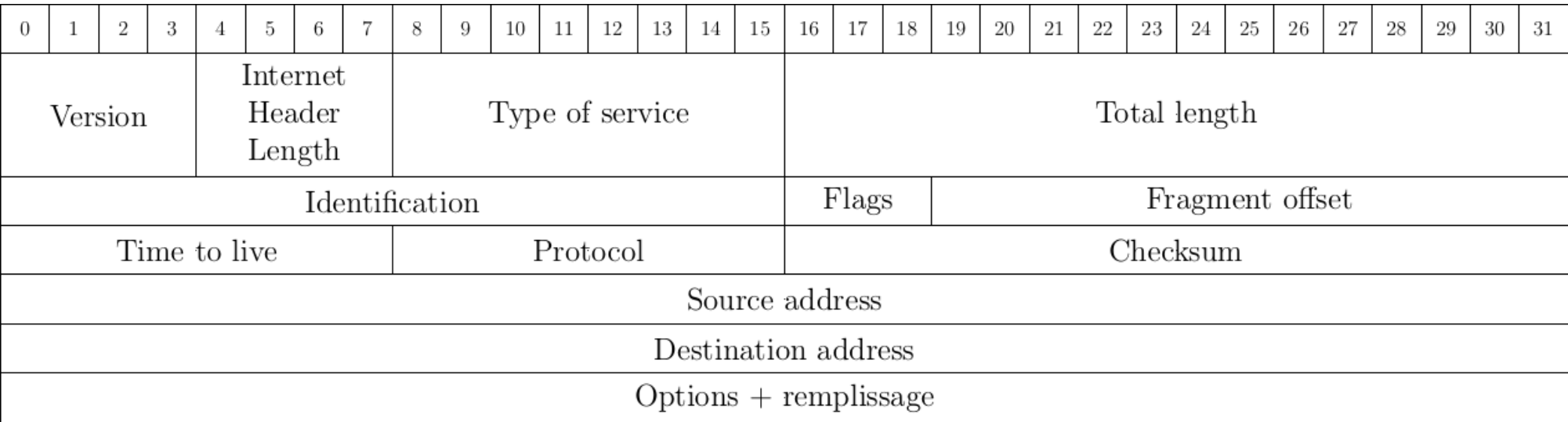
Exemple


192.168.1.1
Pour 132.12.14.56
00:de:ad:be:ef:00
Hello !
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3
16:64:13:37:00:11

Exemple


192.168.1.1
00:de:ad:be:ef:00
Pour: 132.12.14.56
Hello !
147.94.64.36
00:aa:fa:be:ae:17
192.168.1.2
11:22:33:44:55:66
192.168.1.3
16:64:13:37:00:11

DHCP


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
Internet Header Length
Type of service
Total length
Identification
Flags
Fragment offset
Time to live
Protocol
Checksum
Source address
Destination address
Options + remplissage

Config. statique, `ip` vs `ifconfig`

`ip addr show dev eth0`

~~`ifconfig eth0`~~

`ip link set up eth0`

~~`ifconfig eth0 up`~~

`ip link set down eth0`

~~`ifconfig eth0 down`~~

`ip addr add 192.168.54.2/24 dev eth0`

~~`ifconfig eth0 192.168.54.2/24`~~

`ip link set dev eth0 address 00 : 52 : bc : 33 : 25 : a1`

~~`ifconfig eth0 hw ether 00 : 52 : bc : 33 : 25 : a1`~~

Configuration dynamique

Dynamic Host Configuration Protocol

On veut permettre à chaque machine d'obtenir automatiquement :

- une adresse IP ;
- le masque du réseau local ;
- une passerelle par défaut ;
- le nom de domaine ;
- l'adresse d'un serveur DNS pour la résolution.

Serveur DHCP

Les `serveurs DHCP` doivent fournir ces renseignements tout en assurant plusieurs fonctionnalités :

- réutiliser les adresses délaissées ;
- autoriser certaines machines à avoir une IP fixe ;
- se coordonner avec d'éventuels autres serveurs DHCP dédiés au même réseau local.

DHCP est un protocole applicatif encapsulé dans des datagrammes UDP, port serveur 67, port client 68.

Le protocole, RFC 2131

Échange initial standard

- **DHCPDISCOVER** : le client cherche les serveurs DHCP, broadcast.
- **DHCPOFFER** : le serveur propose un bail au client, unicast.
- **DHCPREQUEST** : le client accepte l'offre, broadcast.
- **DHCPACK** : le serveur valide, unicast.

Dans le cas de plusieurs serveurs DHCP, un seul échange arrive à terme, le DHCPREQUEST en broadcast informe les autres serveurs d'un refus.

Renouvellement

Bail DHCP

```
lease 192.168.2.109 {
starts 0 2018/03/11 13 :22 :12 ;
ends 0 2018/03/11 14 :22 :12 ;
cltt 0 2018/03/11 13 :22 :12 ;
binding state active ;
next binding state free ;
hardware ethernet 62 :23 :07 :88 :eb :5c ;
}
lease 192.168.3.109 {
starts 0 2018/03/11 13 :22 :20 ;
ends 0 2018/03/11 14 :22 :20 ;
cltt 0 2018/03/11 13 :22 :20 ;
binding state active ;
next binding state free ;
hardware ethernet 2a :a2 :c2 :6b :f5 :26 ;
}
```

```
lease 192.168.1.109 {
starts 0 2018/03/11 13 :23 :12 ;
ends 0 2018/03/11 13 :23 :52 ;
cltt 0 2018/03/11 13 :23 :12 ;
binding state active ;
next binding state free ;
hardware ethernet 56 :1c :2c :3e :6e :e9 ;
}
lease 192.168.1.109 {
starts 0 2018/03/11 13 :23 :29 ;
ends 0 2018/03/11 13 :24 :09 ;
cltt 0 2018/03/11 13 :23 :29 ;
binding state active ;
next binding state free ;
hardware ethernet 56 :1c :2c :3e :6e :e9 ;
}
```

- DHCPREQUEST : le client demande le renouvellement, broadcast.
- DHCPACK : le serveur valide, unicast.

Le protocole

Autres types de message

- DHCPDECLINE : le client refuse l'adresse IP (e.g. : le client détecte que l'adresse est déjà utilisée).
- DHCPRELEASE : le client met fin au bail de manière anticipée.
- DHCPINFORM : le client demande les paramètres de configuration (pas une adresse IP).
- DHCPNACK : le serveur informe le client de la fin de son bail.

DHCP en pratique

Solution DHCP libre très répandue sous linux : `isc-dhcp-client`, `isc-dhcp-server` et `isc-dhcp-relay`.

Dans labex (sur celene), boxc est un serveur DHCP et fournit une adresse IP à charlie.

Côté client :

- lancement du client avec dhclient
- configuration dans /etc/dhcp/dhclient.conf
- baux dans /var/lib/dhcp/dhclient.leases

DHCP en pratique

Côté serveur

- lancement du serveur avec /etc/init.d/isc-dhcp-server start
- configuration dans /etc/dhcp/dhcpd.conf
- baux dans /var/lib/dhcp/dhcpd.leases

Configuration serveur *isc-dhcp*

`/etc/dhcp/dhcpd.conf` de boxc dans labex

```
subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.50 192.168.0.100 ;
        option routers 192.168.0.1 ;
        option domain-name "green.net" ;
        option domain-name-servers 3.4.5.6 ;
}
```

Configuration serveur *isc-dhcp*

`/etc/dhcp/dhcpd.conf` de server dans labsock

```
  option domain-name "iiia.net" ;
  option domain-name-servers 10.0.1.1 ;

subnet 10.0.0.0 netmask 255.0.0.0 {
        range 10.0.0.1 10.0.0.50 ;

host pc1 { hardware ethernet 6e :5f :98 :37 :0c :07 ; fixed-address pc1.iiia.net ; }
host pc2 { hardware ethernet 2e :18 :cc :d4 :8c :e7 ; fixed-address pc2.iiia.net ; }
host pc3 { hardware ethernet 2e :fe :a2 :81 :23 :ce ; fixed-address pc3.iiia.net ; }
host pc4 { hardware ethernet ce :53 :0c :0c :ef :61 ; fixed-address pc4.iiia.net ; }
}
```

Dnsmasq

Sous linux, l'utilitaire `dnsmasq` fournit :

- serveur DHCP ;
- relais DNS ;
- serveur BOOTP, protocole d'amorçage ;
- serveur TFTP, protocole de transfert de fichier utilisé au démarrage (e.g. démarrer depuis une carte réseau).

Configuration serveur *dnsmasq*

Dans `/etc/dnsmasq.conf`

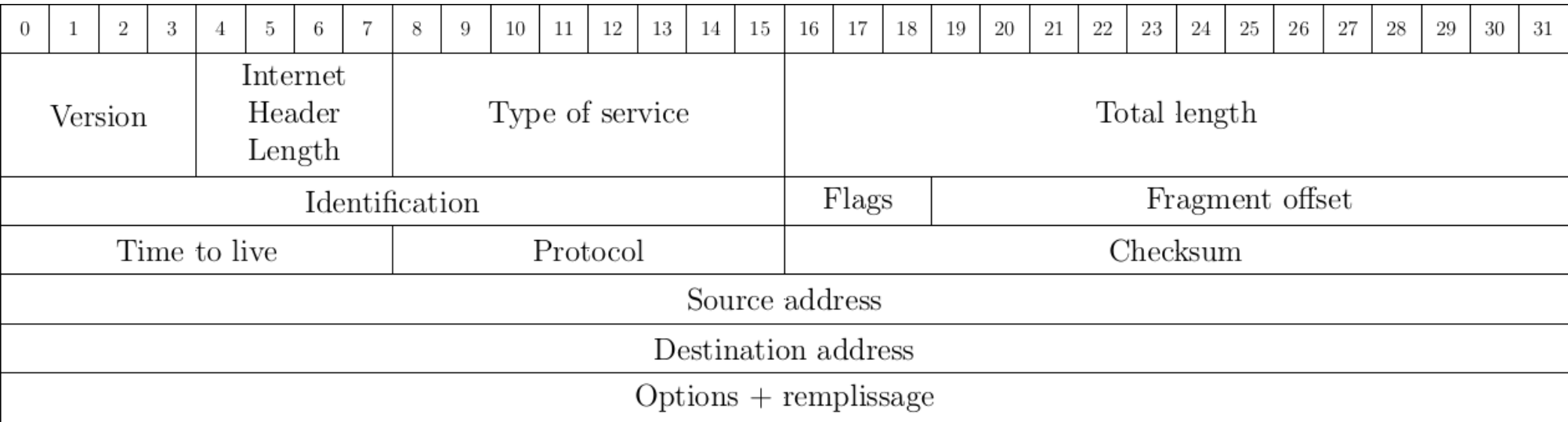
```
interface=eth0          # pour limiter l'écoute DHCP à cette interface
domain=green.net


dhcp-authoritative      # si un seul serveur DHCP
dhcp-leasefile=/tmp/dhcp.leases



# Plage DHCP
dhcp-range=192.168.0.50,192.168.0.100,12h
# Netmask
dhcp-option=1,255.255.255.0
# Route
dhcp-option=3,192.168.0.1
# Serveur DNS
dhcp-option=6,3.4.5.6
# IP fixe
dhcp-host=6e :5f :98 :37 :0c :07,192.168.0.27
```

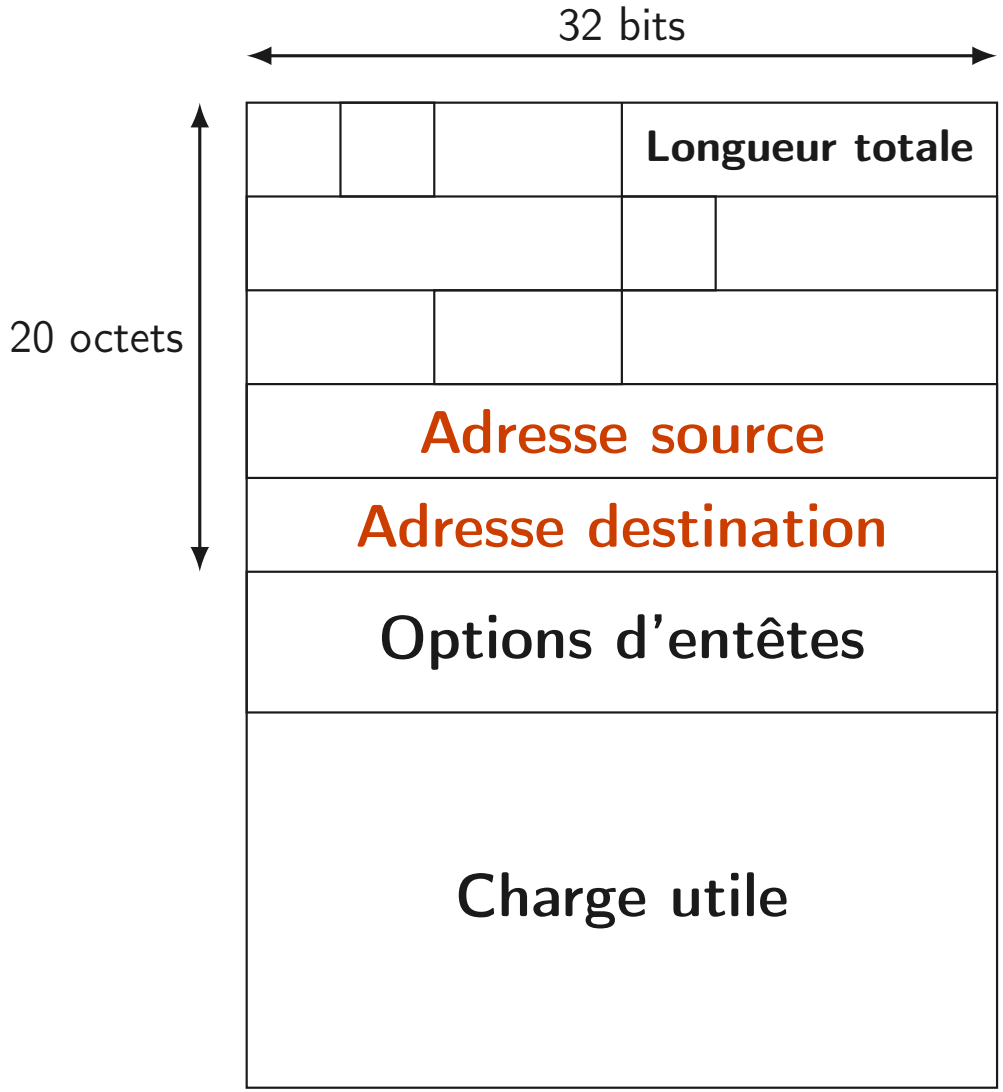
Paquets IP


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
Internet Header Length
Type of service
Total length
Identification
Flags
Fragment offset
Time to live
Protocol
Checksum
Source address
Destination address
Options + remplissage

Donnée unitaire

- Le paquet IP est la structure de l'envoi en couche réseau.
- Il est encapsulé dans une trame ethernet (ou wifi).
- Il contient souvent (Chap 5) un datagramme UDP ou un segment TCP.
- Il a une adresse source et une adresse destination.

Paquet IP : contenu



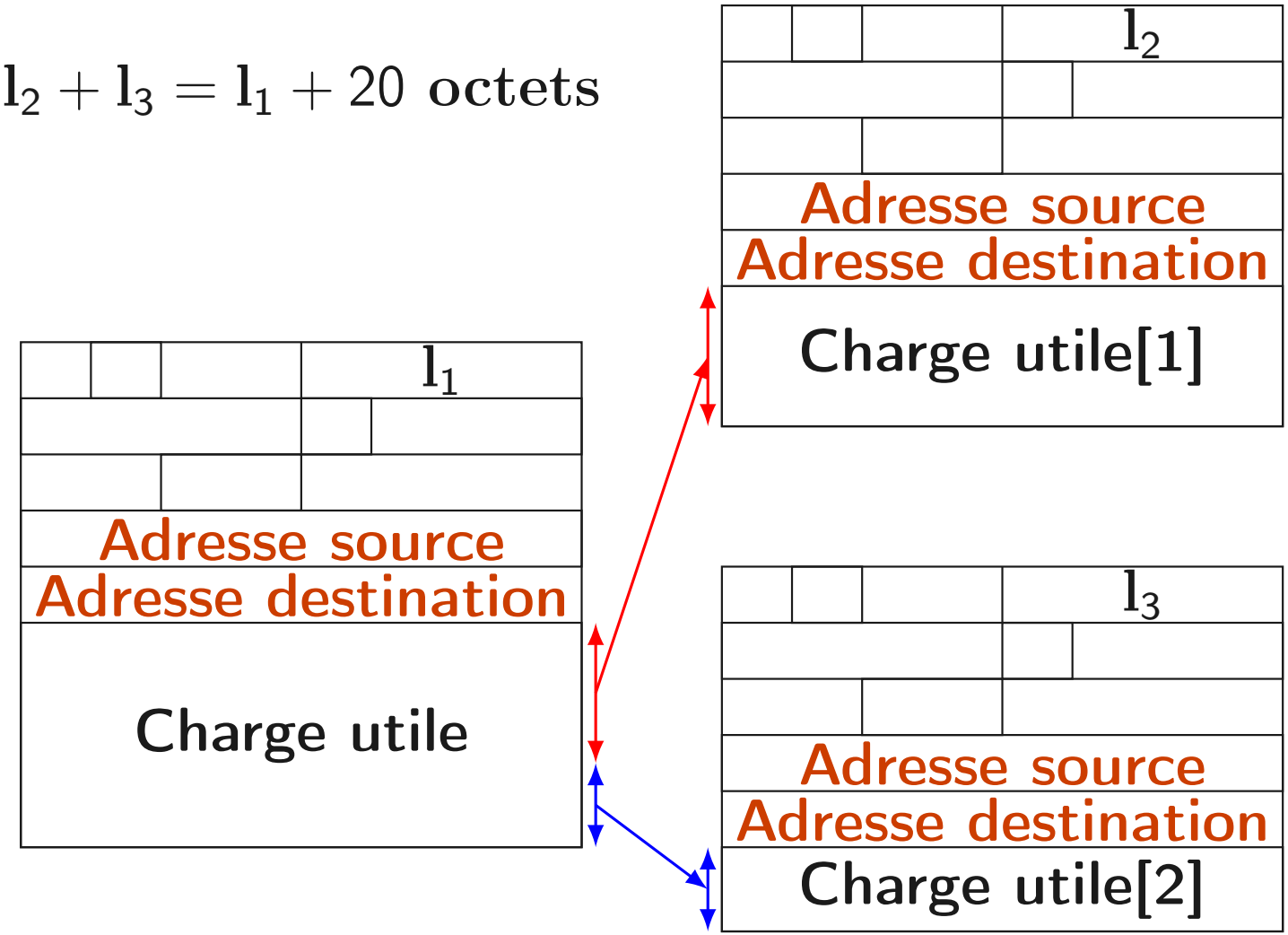
Longueur totale encodée sur 16 bits : maximum 64Ko.

Paquet IP : fragmentation

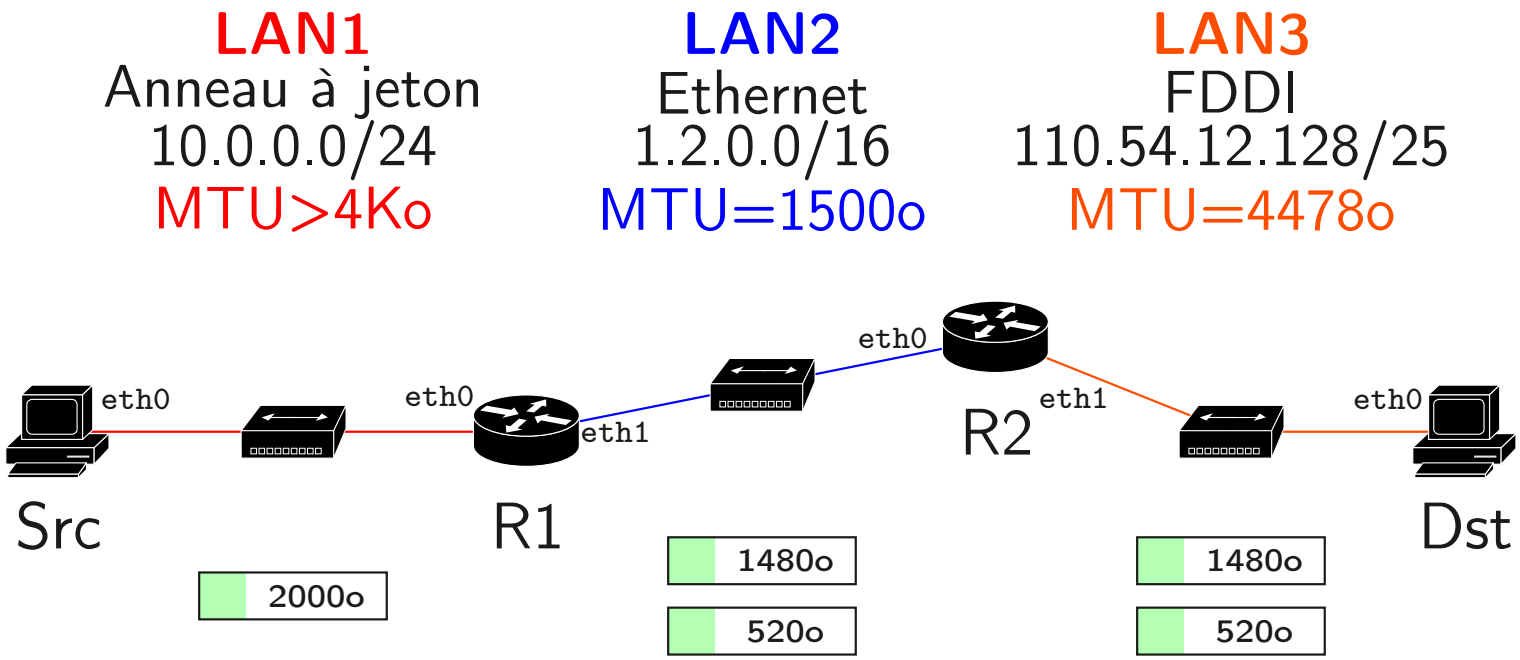
- Les trames ethernet ont une MTU de 1500 octets : trop petit pour un paquet IP a priori.
- Intérêt en couches basses : éviter la monopolisation du médium, diminuer le risque d'erreur.
- Pour transmettre un paquet IP, il faut le `fragmenter`.
- Deux options :
  - chaque routeur adapte les paquets à la MTU du médium emprunté (IPv4) ;
  - fragmentation aux extrémités uniquement (IPv6).

Fragmentation en pratique

$l_2 + l_3 = l_1 + 20$ octets

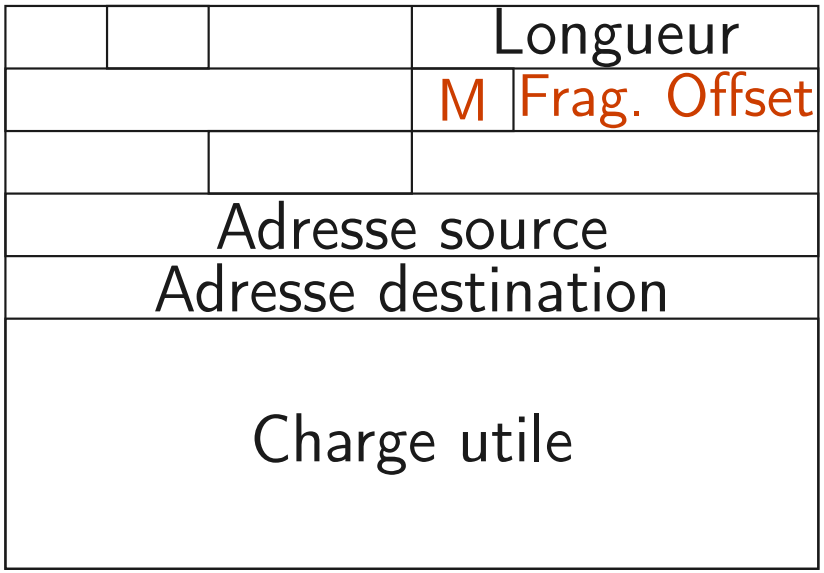


Fragmentation



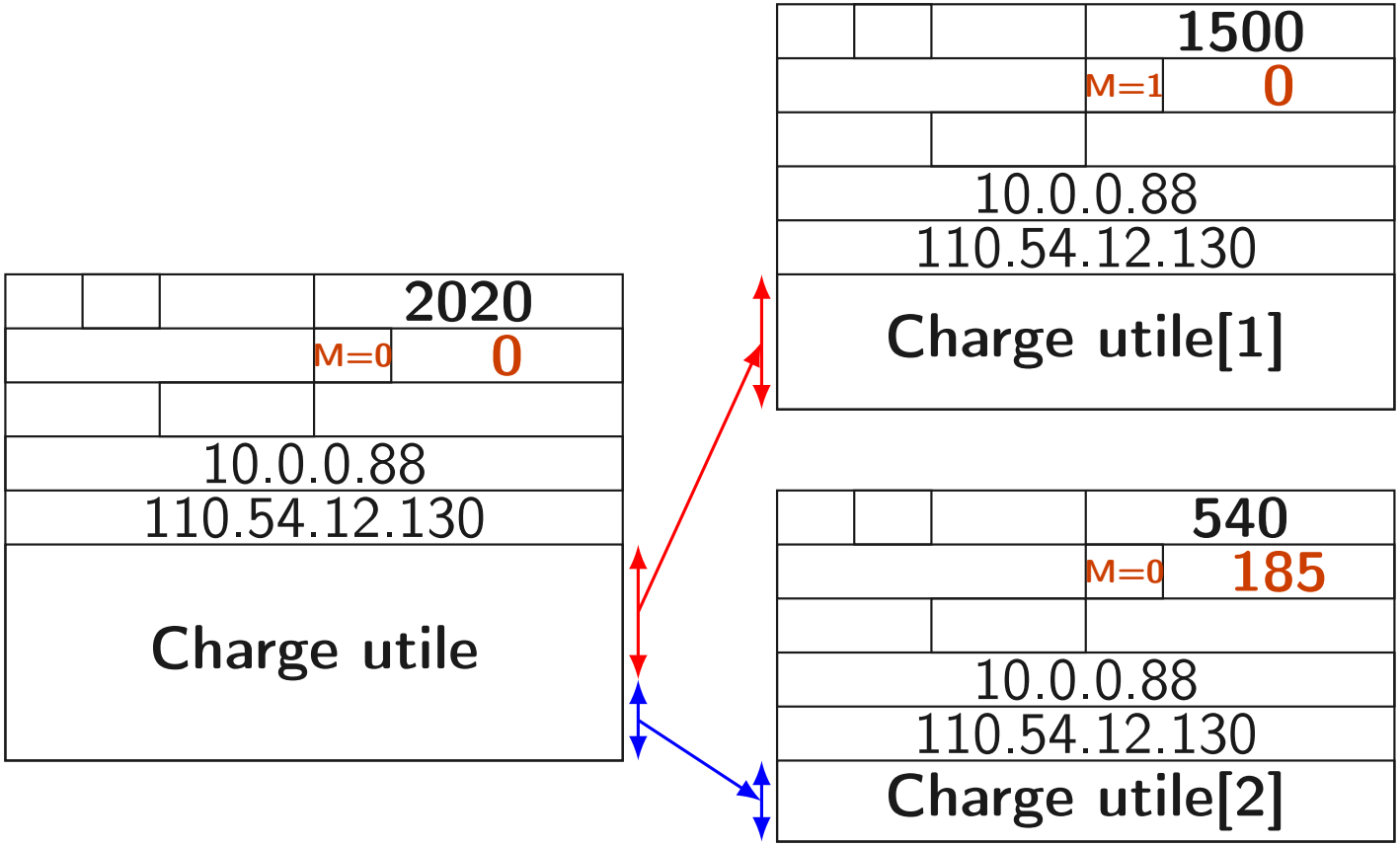
Un paquet de 2000 octets est envoyé par la source, fragmenté par $R1$ en deux paquets transmis par $R2$. Finalement, la destination assemble les paquets.

Dans un paquet IP



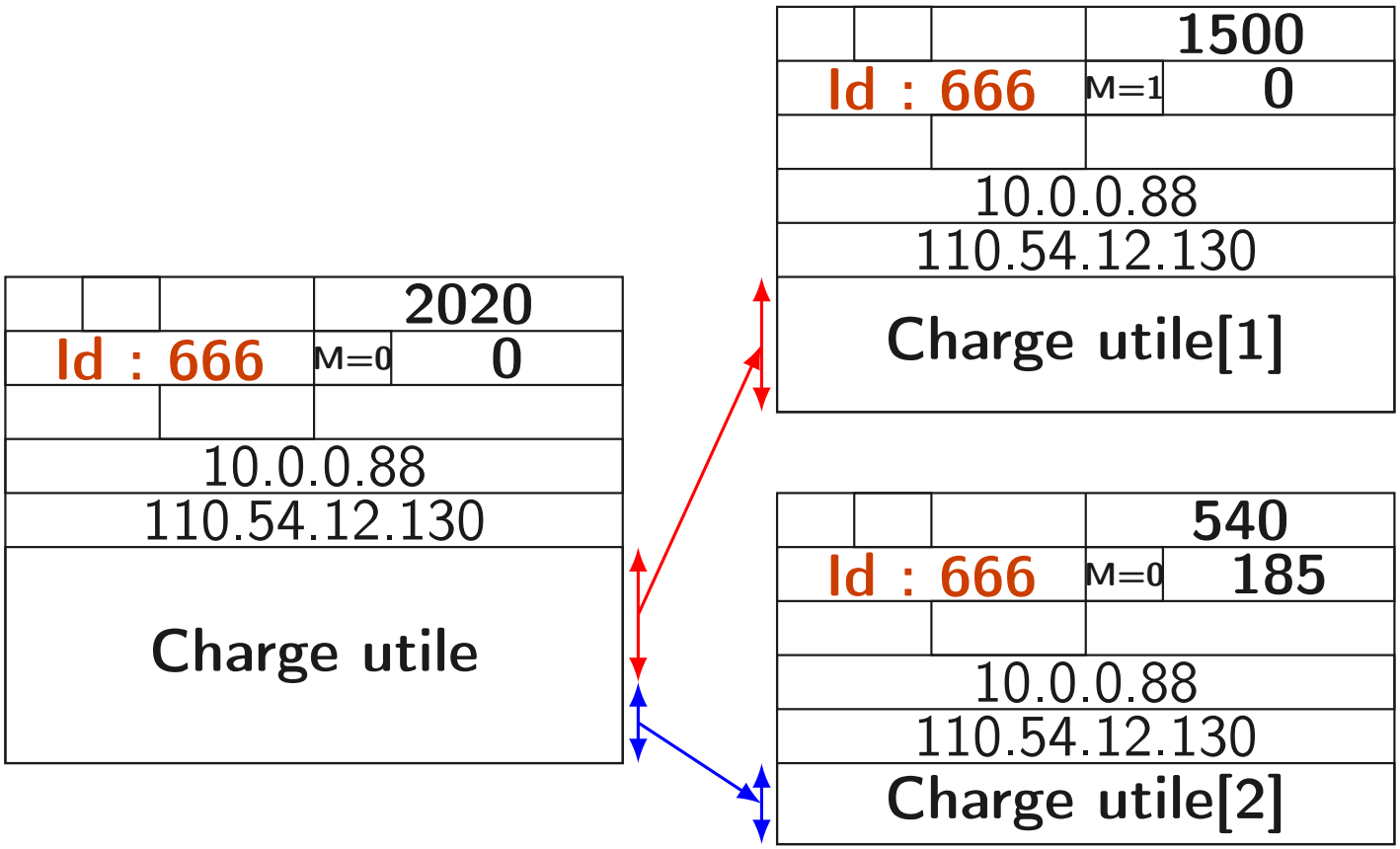
- La longueur est celle du fragment.
- Le décalage (offset) est le nombre de blocs de 8 octets passés avant le fragment.
- M (More bits) est un bit qui vaut 1 sauf pour le dernier fragment.

Fragmentation, exemple



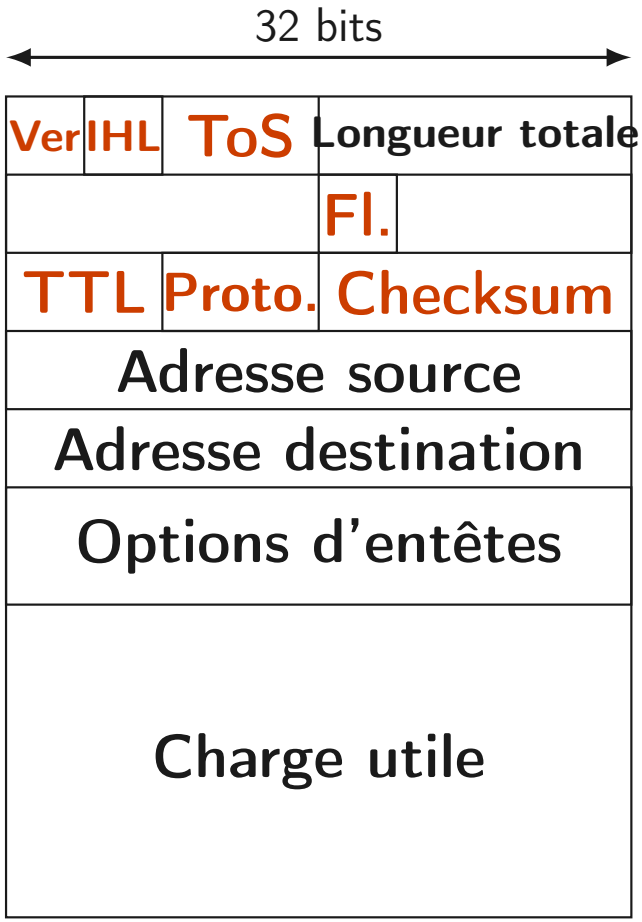
Un paquet de 2000 octets fragmenté en 1480 et 520 de charge utile (payload).

Fragmentation, exemple



Le champ identification est choisi par l'émetteur : un identifiant n'est réutilisé par un hôte qu'après une période de temps suffisante pour éviter des confusions.

Format d'un paquet IP



- **Ver** : Version du protocole (4 ou 6)
- **IHL** : Internet Header Length (entre 20 et 64)
- **ToS** : Type of Service (surtout pour UDP : faible latence ou fiabilité haute ou . . .)
- **Fl.** : Flags (notamment **More** ou **Don't fragment**) sur 3 bits
- **TTL** ; Time To Live (en nombre de sauts)
- **Proto.** : Protocol (e.g. UDP, TCP)
- **Checksum** : uniquement sur l'entête

Fiabilité

- Un paquet IP peut être transmis avec erreur : aucun contrôle sur le contenu.
- L'entête est contrôlée (`checksum`).
- Le champ `ToS` peut servir à demander un délai court dans certains cas et une fiabilité plus grande.
- Le `TTL` est décrémenté par chaque routeur relayant le paquet.
- S'il atteint 0, le paquet est détruit et un message d'erreur (`protocole ICMP`) est envoyé à l'émetteur.

ICMP

Internet Control Message Protocole

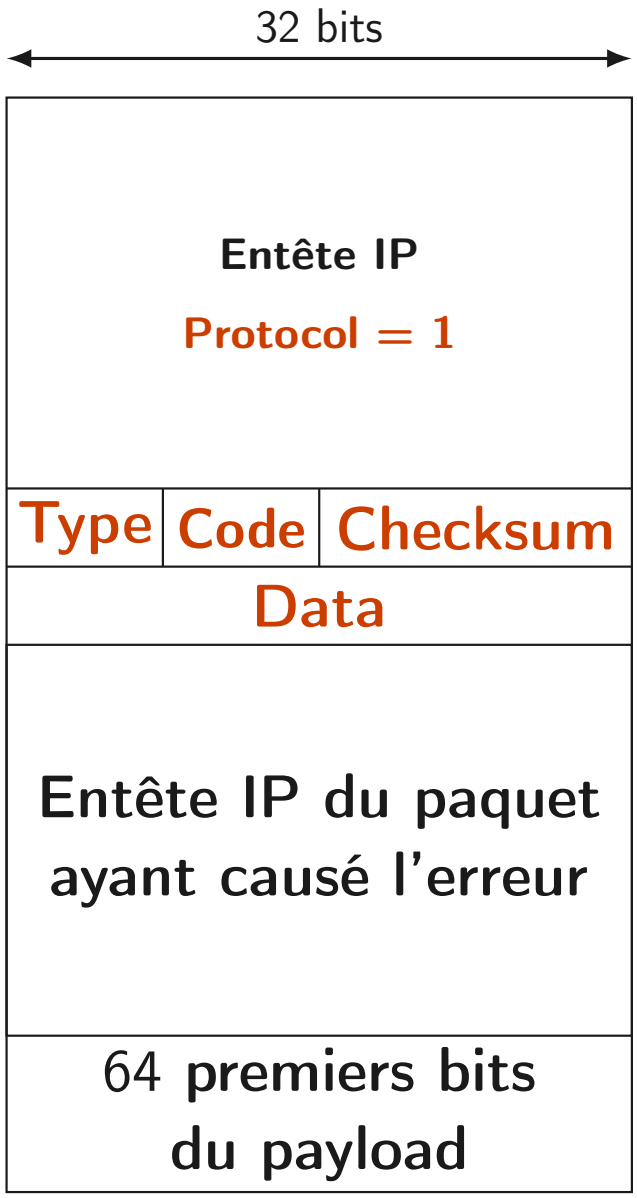
Protocole de la couche réseau :

- Erreurs : destinataire inaccessible, entête erronée...
- Diagnostics : demande de ralentissement des envois en cas de congestion, information de routage inefficace...

Des exemples :

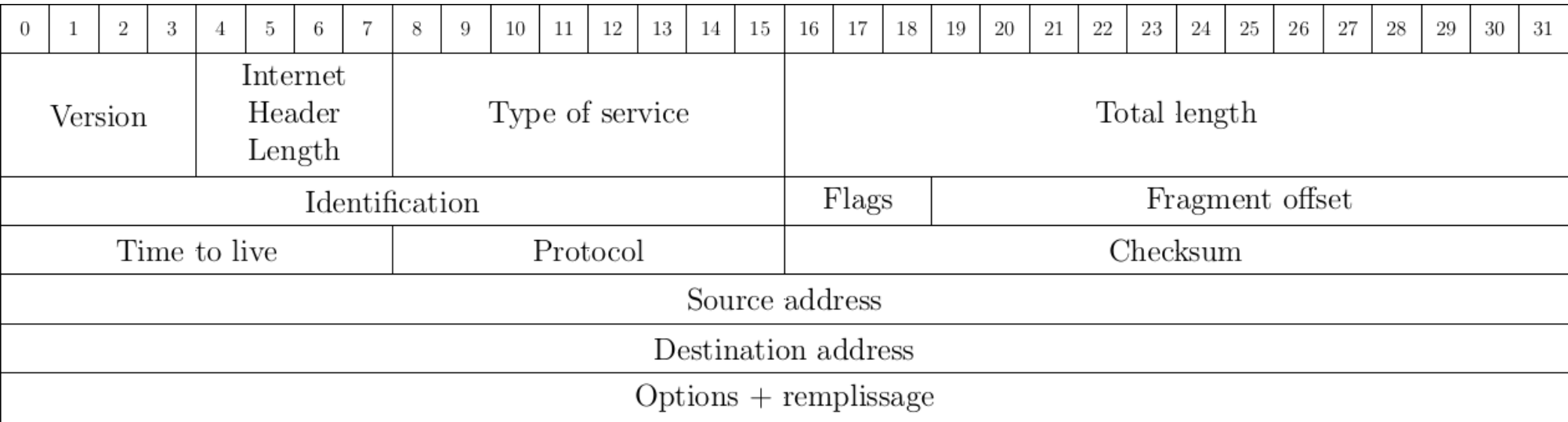
- Ping : ECHO request et reply
- Destination unreachable
- Traceroute utilise ICMP : suite de paquets IP avec des TTL incrémentés depuis 1. Le message ICMP correspondant à un TTL initialisé à i donne le ième intermédiaire.
- Fragmentation needed : pour estimer la MTU minimale d'un chemin

Paquet ICMP



- **Type** et **Code** : type et code de message ICMP (ECHO, Dest. Unreachable, TTL exceeded...)
- **Checksum** : sur le message ICMP complet

Routage


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
Internet Header Length
Type of service
Total length
Identification
Flags
Fragment offset
Time to live
Protocol
Checksum
Source address
Destination address
Options + remplissage

À qui envoyer un paquet ?

Lorsqu'un hôte souhaite envoyer un paquet :

- si le destinataire est sur le même réseau local (adresse réseau, masque), envoi direct (résolution ARP, encapsulation dans une trame).
- sinon, il l'envoie à une `passerelle`, i.e. un routeur situé dans le réseau local qui sert de premier intermédiaire : envoi à la passerelle (résolution ARP, encapsulation dans une trame).

- Chaque hôte doit pouvoir trouver à qui envoyer le paquet : `passerelle par défaut`.
- Potentiellement des passerelles différentes selon les destinations.

Router

Un routeur est une machine possédant des interfaces dans plusieurs réseaux locaux :

- c'est le lien entre LAN.
- pour chaque paquet entrant, il décide de le transmettre ou non et sur quel interface.
- il possède un buffer en entrée contenant les paquets en attente de traitement ;
- et un buffer en sortie contenant les paquets à envoyer.
- si le buffer d'entrée se remplit plus vite que le routeur ne traite les paquets : `congestion`.
- si le buffer de sortie ne se vide pas assez vite (médium trop occupé) : `congestion`.

Inside a router

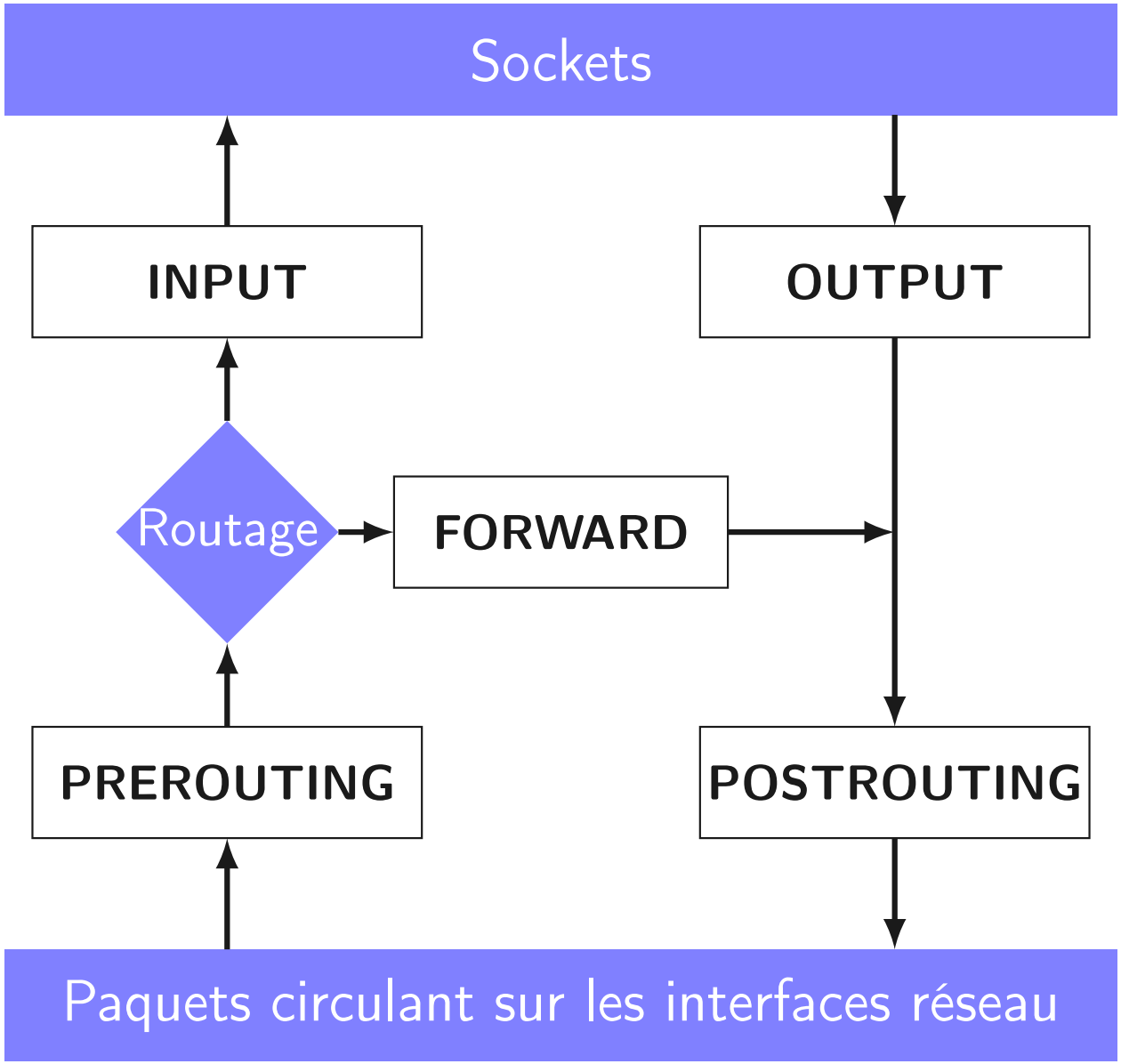


Table de routage

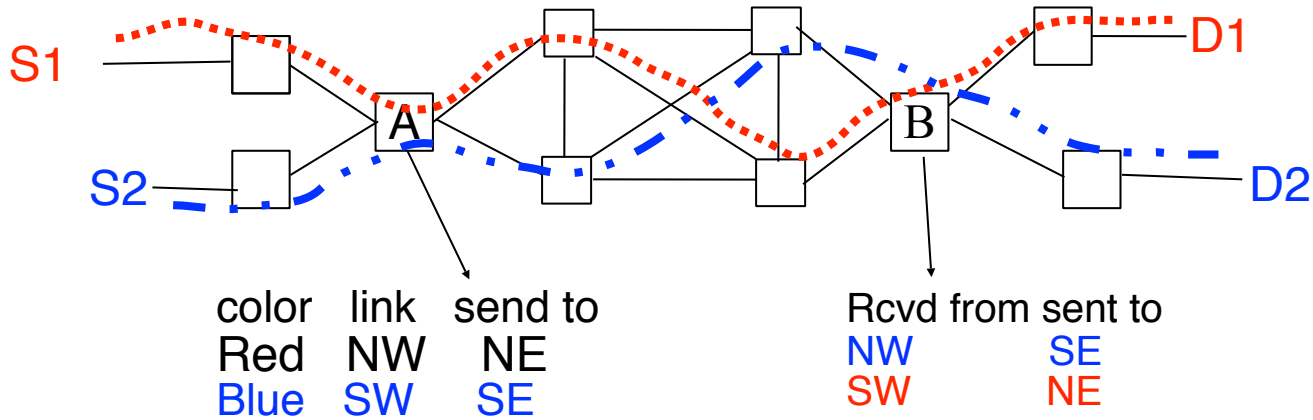
- Pour prendre ses décisions, chaque routeur possède une `table de routage`.
- À chaque réseau correspond une direction (un autre routeur ou la cible si le routeur a une interface dans le LAN destination).
- En général une valeur par défaut : 0.0.0.0.
- La table est remplie en commençant par les entrées `les plus spécifiques` : dès qu'on trouve une entrée qui correspond, on s'arrête.

Remplir la table

Deux options de routage :

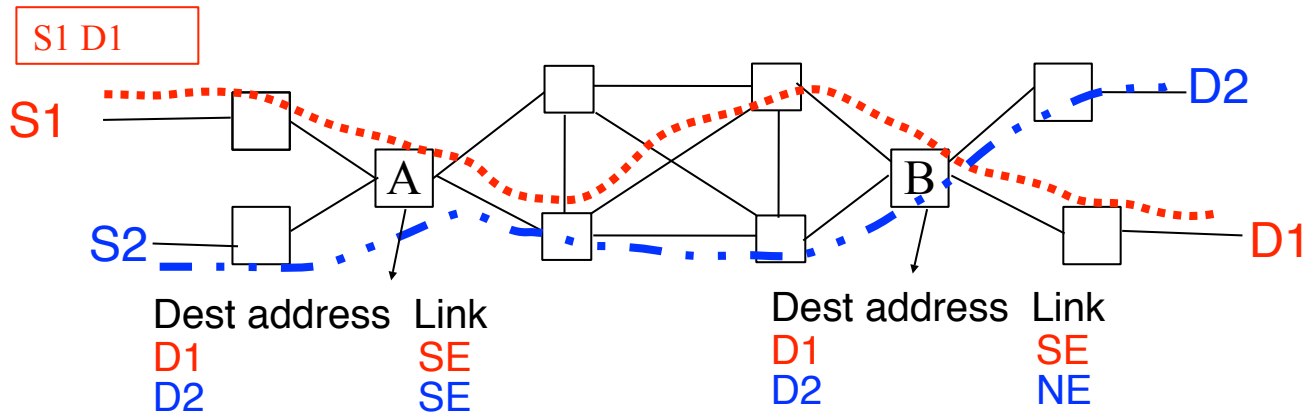
- `Commutation de circuits`. Le chemin des paquets est décidé à l'avance, tous les paquets empruntent le même chemin (`MultiProtocol Label Switching` couche 2.5).
- `Commutation de paquets`. Chaque paquet est routé indépendamment (la table peut changer au cours du temps), notamment le protocole IP.

Commutation de circuits



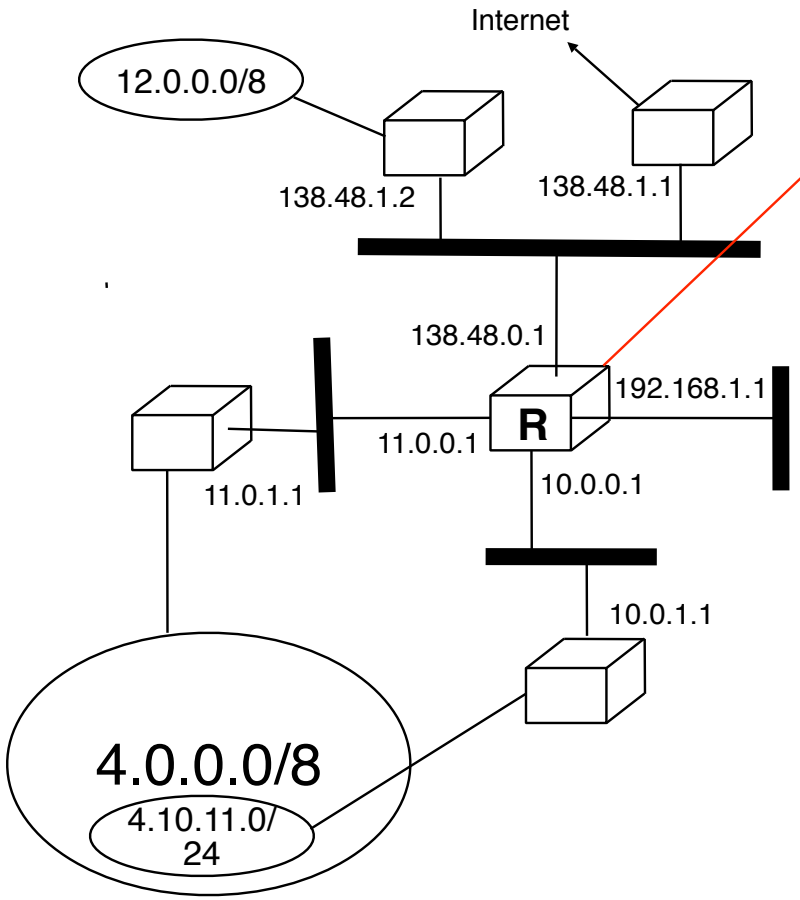
- La table de routage contient des `étiquettes` d'entrée et de sortie : la lecture de l'étiquette permet de choisir l'étiquette adaptée au chemin et l'interface de sortie.
- Garanties de débit et de délai.

Commutation de paquets



- Pour chaque réseau cible, la table de routage contient l'adresse du routeur à qui transmettre le paquet.
- Chaque ligne de la table contient l'adresse du réseau cible avec son masque, l'adresse du routeur et l'interface de sortie.

Exemple



Local addresses
11.0.0.1 ; 138.48.0.1 ; 192.168.1.1 ; 10.0.0.1

Routing table
138.48.0.0/16 INorth]
11.0.0.0/8 [West]
192.168.1.0/24 [East]
10.0.0.0/8 [South]
4.0.0.0/8 via 11.0.1.1 [West]
4.10.11.0/24 via 10.0.1.1 [South]
12.0.0.0/8 via 138.48.1.2 [North]
0.0.0.0/0 via 138.48.1.1 [North]

Remplir la table

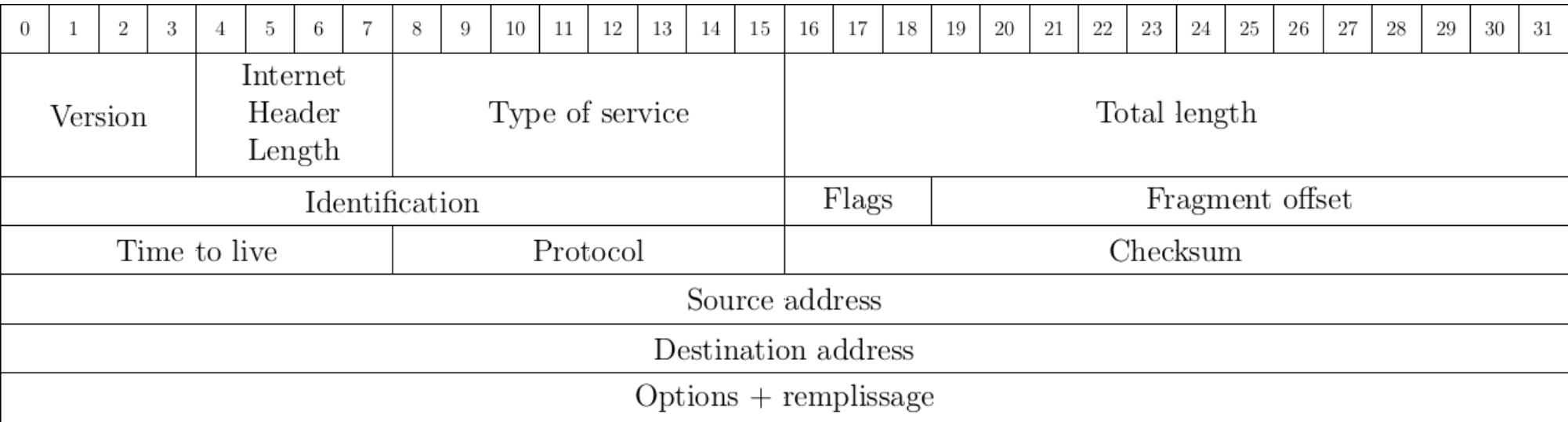
Plusieurs options pour remplir la table :

(1) table `statique`.
Commandes :

- ip route add 114.1.210.0/24 via 10.1.2.3
- ip route add default via 10.1.2.3

(2) table `dynamique`.
Algorithmes de routage distribués.

NAT


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Version
Internet Header Length
Type of service
Total length
Identification
Flags
Fragment offset
Time to live
Protocol
Checksum
Source address
Destination address
Options + remplissage

Adresses IP privées ?

Besoin d'un traducteur

Plusieurs blocs d'adresses privées :

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 169.254.0.0/16

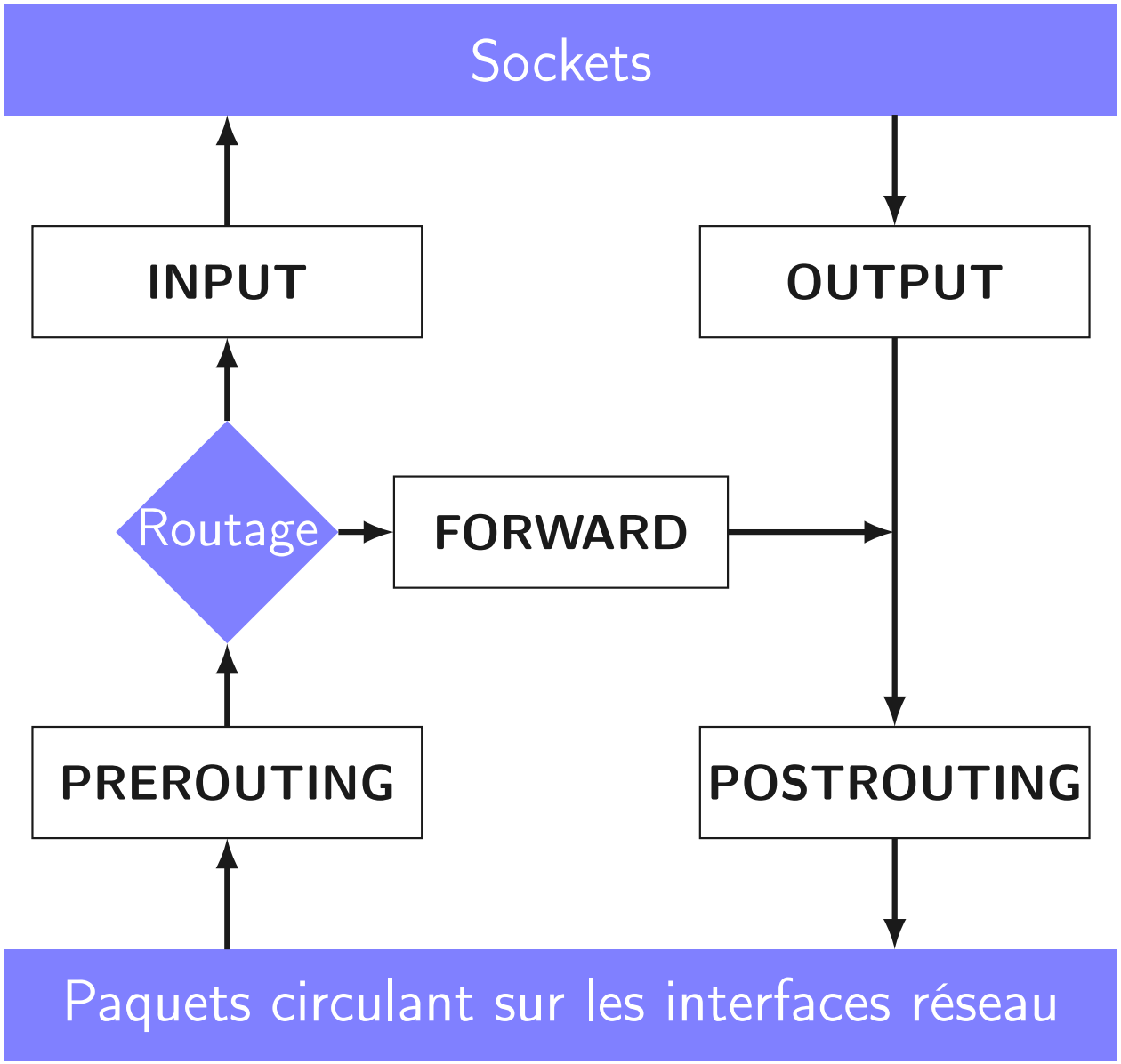
Utiles pour économiser des adresses IPv4 !
Comment utiliser ces adresses sur internet ?
Exemple : le FAI ne fournit qu'une seule adresse publique et la box fait l'interface avec le réseau local.

Address Translation

Plusieurs options, mais dans tous les cas :

- il faut réécrire les adresses privées ;
- en utilisant la ou les adresse(-s) publique(-s) à disposition ;
- changer l'adresse source des paquets sortants ;
- changer l'adresse destination des paquets entrants.

Cas d'un routeur



Basic NAT

Network Address Translation

Si l'on dispose d'autant d'adresses privées que publiques :

- table de correspondance $1-1$;
- l'adresse privée est remplacée par l'adresse publique dans les paquets sortants ;
- l'adresse publique est remplacée par l'adresse privée dans les paquets entrants ;
- intérêt ?

NAT - PAT

Port Address Translation

Si l'on dispose de moins d'adresses publiques, par exemple une seule :

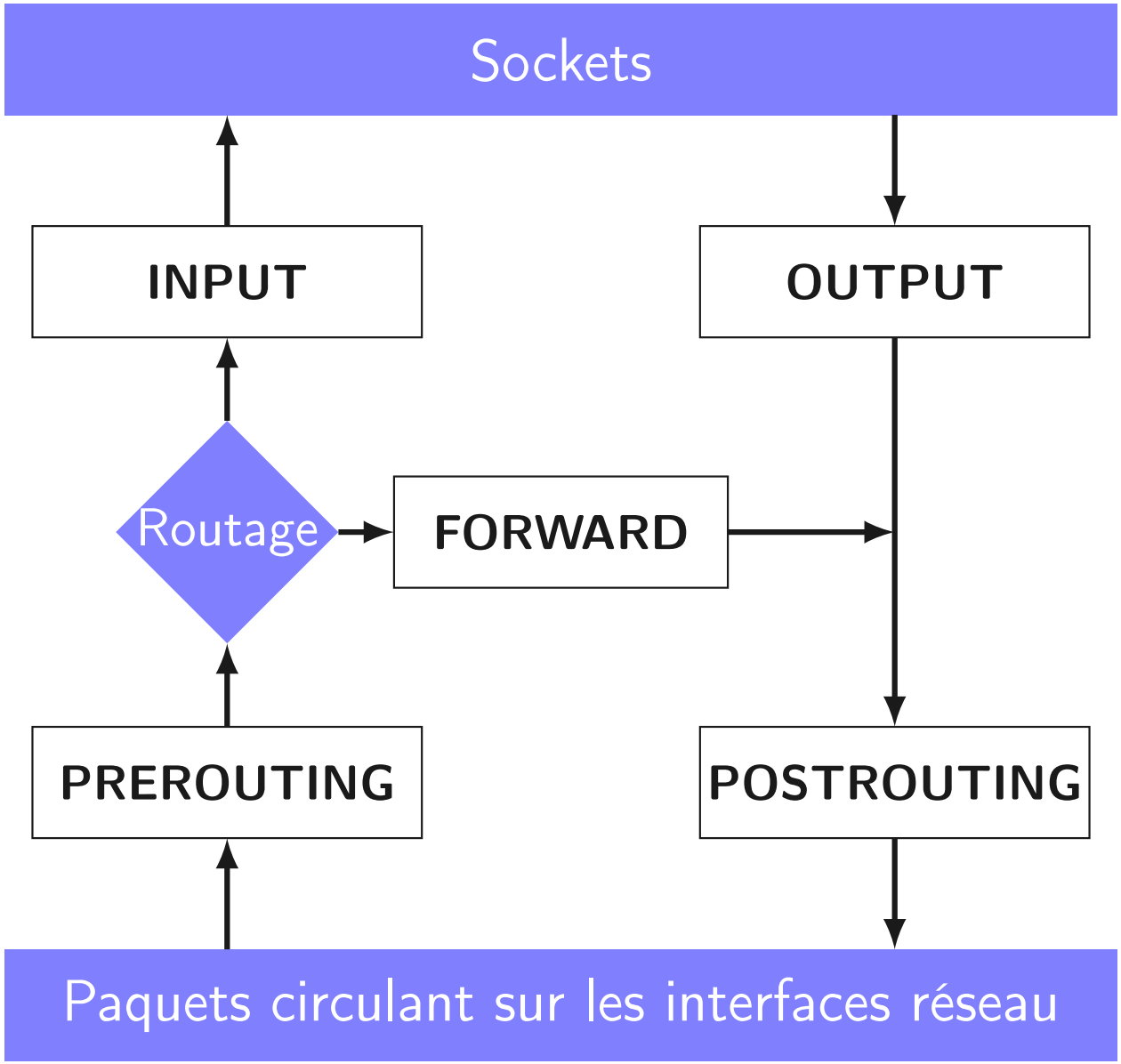
- la passerelle transforme l'adresse privée en l'adresse publique dans les paquets sortants en choisissant un numéro de port approprié ;
- processus inverse pour les paquets entrants : le numéro de port du paquet permet à la passerelle de choisir l'adresse privée de l'hôte concerné ;
- la passerelle maintient une table d'association.
- Quid des protocoles sans port ?

NAT en pratique : iptables

Utilitaire linux (`http://netfilter.org`).

- Permet de faire du filtrage réseau au niveau du noyau linux.
- Implémente par exemple NAT mais aussi des pare-feux.
- paquet `iptables`.
- Fonctionnement par tables qui regroupent des ensembles de règles, e.g. : NAT.

Tables iptables



Commandes iptables

```
                  -A
iptables -t TABLE -I CHAINE REGLE
                  -D
```

- TABLE : contexte, par exemple nat, filter,...
- CHAINE : chaque table est une collection de chaînes.
- REGLE : chaque chaîne est une liste ordonnée de règles (conditions + cible).
- -A/I/D pour ajouter, insérer (donner position), supprimer.

Exemples SNAT/DNAT

```
iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to
1.2.3.54
```

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
                              -j DNAT --to 10.0.0.2:8080
```

Construire un filtre

Possibilité de :

- créer ou supprimer des tables, des chaînes.
- envoyer vers une cible -j CIBLE, en particulier ACCEPT, REJECT (avec message d'erreur ICMP), DROP (sans message d'erreur) ou une autre chaîne.
- utiliser des conditions pour appliquer la règle sur l'adresse ou le port (src ou dst), le protocole, l'interface (entrée ou sortie), le type de message,...

Exemples SNAT

NAT source

iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to 1.2.3.54

- Remplacer l'adresse source (réseau privé) par l'adresse passée en paramètre.
- En POSTROUTING pour connaître l'adresse IP de l'interface de sortie (e.g. routeur à 3 interfaces).
- La box (ou routeur NAT) écrit dans une table l'association effectuée, si un paquet réponse arrive sur le port choisi, l'adresse destination sera substituée.

Exemples DNAT

NAT destination

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
                              -j DNAT --to 10.0.0.2:8080
```

- Remplacer l'adresse de destination (adresse publique de la box) par l'adresse dans le réseau privé.
- Utile par exemple pour héberger un serveur web dans un réseau privé (derrière un pare-feu).
- Un paquet est transmis uniquement si il contient du TCP sur le port 80.

Pare-feu

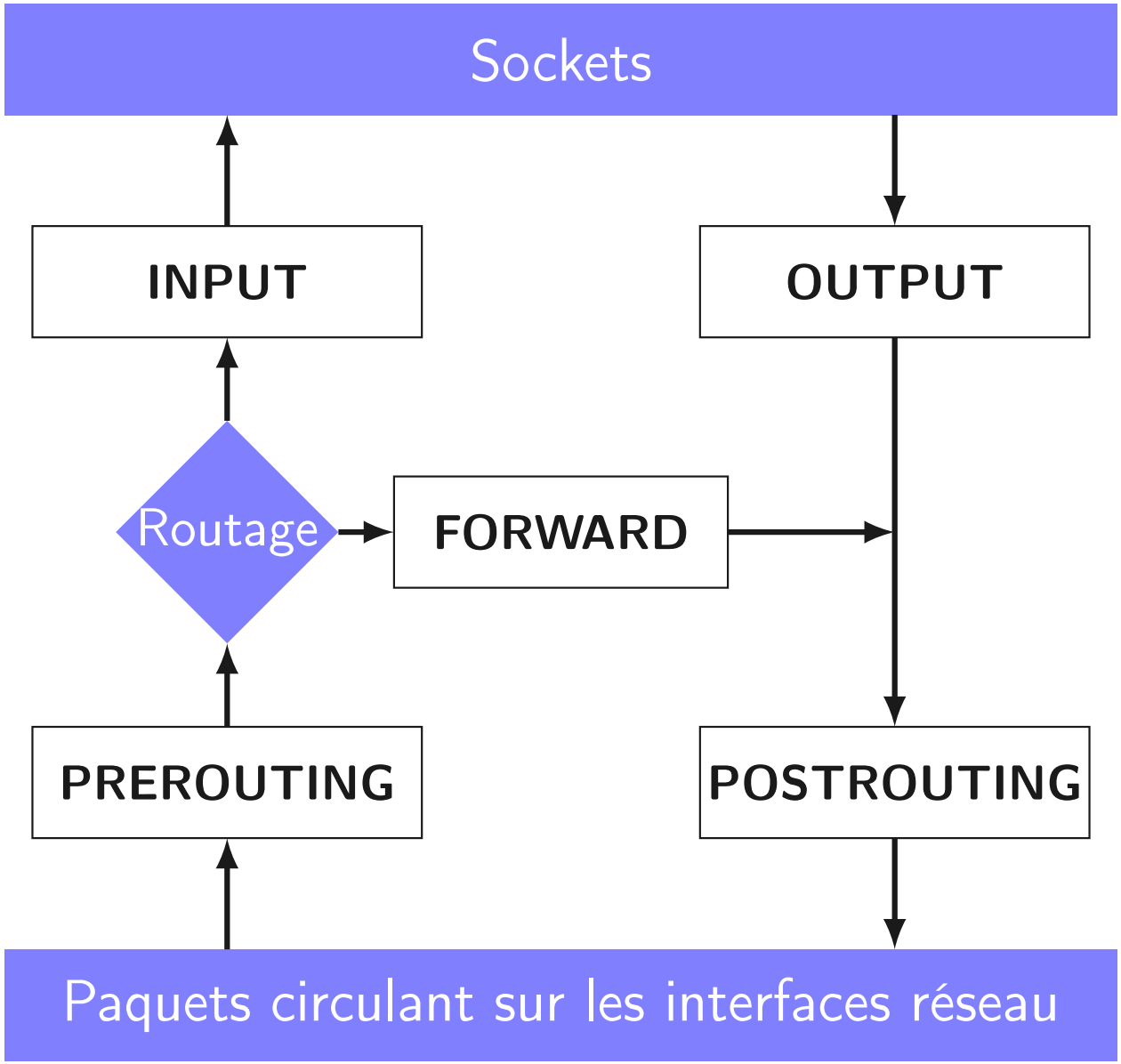
Table `filter`

Plusieurs tables existent, en particulier `filter` (table par défaut) pour configurer le pare-feu :

- Chaîne `INPUT` pour les paquets entrants (après décision de routage).
- Chaîne `OUTPUT` pour les paquets sortants.
- Chaîne `FORWARD` pour les paquets à transférer (après décision de routage).

Politique de contrôle et sécurité. La décision du sort des paquets est prise après lecture des entêtes de couche 2, 3 et 4.

Pare-feu avec iptables



Autres usages

SSH only (TCP 22)

! pour la négation de la condition.

```
iptables -t filter -A INPUT -p  !tcp -j DROP

iptables -t filter -A INPUT --dport  !22 -j DROP
```

Autres usages

DMZ

Configurer un routeur/pare-feu à 3 pattes avec :

- eth0 vers l'extérieur (internet) ;
- eth1 vers la DMZ qui contient un serveur HTTP ;
- eth2 vers le réseau d'entreprise qui contient un serveur PostgreSQL accessible au serveur HTTP sur le port TCP 5432.

On supposera que l'extérieur n'accède qu'au serveur HTTP, le réseau d'entreprise à tout l'internet et la DMZ à l'extérieur et au serveur PostgreSQL uniquement.

Autres usages

DMZ

Préparer le terrain.

```
iptables -t filter -F FORWARD
iptables -t filter -F INPUT
iptables -t filter -F OUTPUT
iptables -t nat -F PREROUTING
iptables -t nat -F POSTROUTING
```

Autres usages

DMZ

Règles par défaut

```
iptables -t filter -P FORWARD DROP
iptables -t filter -P INPUT DROP
iptables -t filter -P OUTPUT DROP
```

Autres usages

DMZ

Traffic local.

```
iptables -t filter -A INPUT -i lo -j ACCEPT
iptables -t filter -A OUTPUT -o lo -j ACCEPT
```

Autres usages

DMZ

Autoriser les connections SSH sur le routeur.

```
iptables -t filter -A INPUT -i eth2 -p tcp --dport 22 -j ACCEPT
iptables -t filter -A OUTPUT -o eth2 -p tcp --sport 22 -j ACCEPT
```

Autres usages

DMZ

NAT source : le LAN accède à internet.

```
iptables -t nat -A POSTROUTING -s 192.168.20.0/24 -o eth0 -j SNAT
--to 1.2.3.4
```

```
iptables -A FORWARD -i eth2 -o eth0 -j ACCEPT
```

```
iptables -A FORWARD -i eth0 -o eth2 -j ACCEPT
```

Autres usages

DMZ

NAT destination : internet peut se connecter au serveur web.

```
iptables -t nat -A PREROUTING -d 1.2.3.4 -p tcp --dport 80 -j DNAT
--to-destination 192.168.10.2:80
```

```
iptables -t filter -A FORWARD -i eth0 -o eth1 -p tcp --dport 80 -m state
--state NEW,ESTABLISHED -d 192.168.10.2 -j ACCEPT
```

```
iptables -t filter -A FORWARD -i eth1 -o eth0 -p tcp --sport 80 -m state
--state ESTABLISHED -s 192.168.10.2 -j ACCEPT
```

Autres usages

DMZ

Le serveur web peut interroger le serveur SQL.

```
iptables -A FORWARD -i eth1 -o eth2 -p tcp -s 192.168.10.2 --dport 5432
-m state --state NEW,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -i eth2 -o eth1 -p tcp -d 192.168.10.2 --sport 5432
-m state --state ESTABLISHED -j ACCEPT
```

Autres usages

DMZ

Le LAN peut demander la page web.

```
iptables -A FORWARD -i eth2 -o eth1 -p tcp -d 192.168.10.2 --dport 80
-m state --state NEW,ESTABLISHED -j ACCEPT
```

```
iptables -A FORWARD -i eth1 -o eth2 -p tcp -s 192.168.10.2 --sport 80 -m
state --state ESTABLISHED -j ACCEPT
```

Droits

`http://inl.info.ucl.ac.be/CNP3`

Merci à Nicolas Ollinger.