TCP/IP

Pascal Malterre

pascal.malterre@cea.fr

Introduction

- Interconnexion d'équipements informatiques
  - Les équipements peuvent être des composants électroniques (unité centrale d'un PC), des périphériques (imprimantes), des ordinateurs, etc.
  - Le support de communication peut être le fil électrique, le réseau téléphonique, les ondes radio, etc.
  - On doit gérer l'attribution du support, le temps de parole, etc.
  - Traitement des erreurs (restransmissions, auto-corrections, etc.)
- Mot-clé : hétérogénéïté

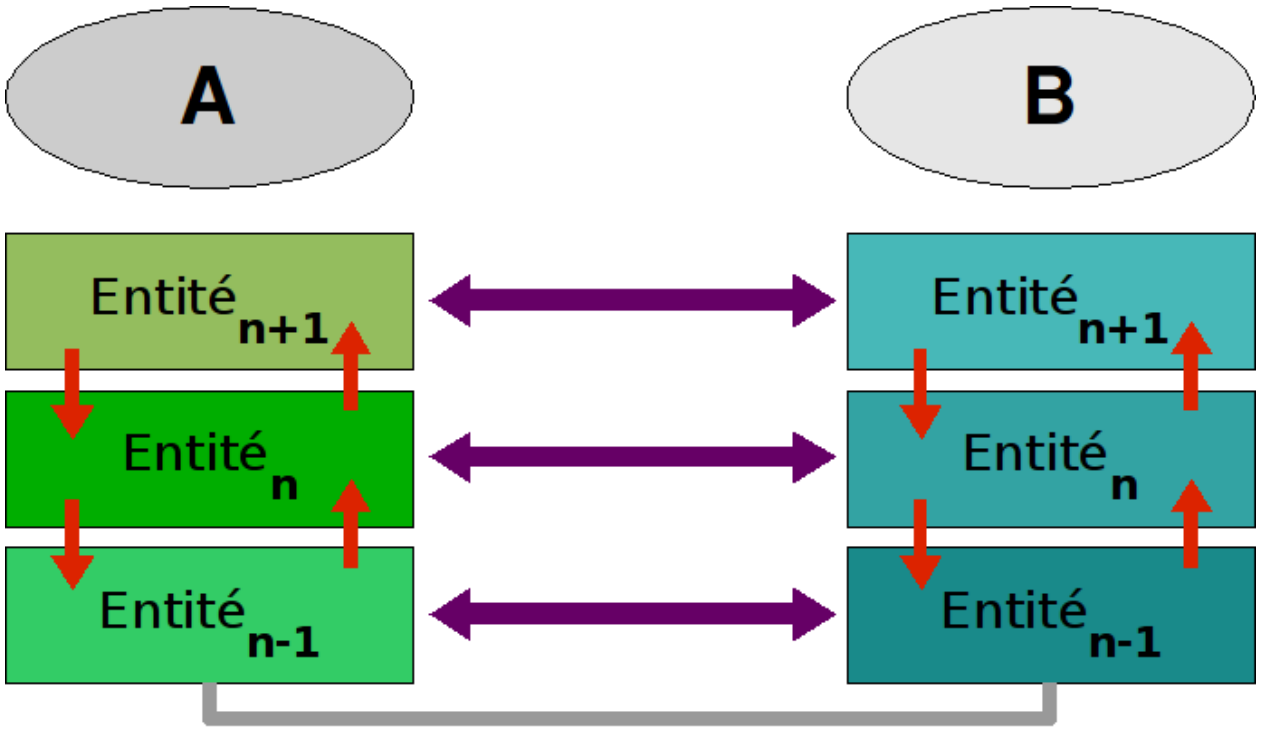
Processus de normalisation

- Normes de fait vs. normes de droit
- L'ISO (International Organization for Standards) regroupe les organisations de normalisation nationaux : ANSI (us), AFNOR (France), etc.
- Pour les télécommunications : UIT/T (ex CCITT), ETSI (European Telecommunications Standards Institute)
- Les normes de l'Internet sont appelées RFC (Request For Comments) et sont élaborées par l'IETF (Internet Engineering Task Force)
  - Les RFCs sont à la fois des normes de fait (les géants de l'informatique participent à leurs élaborations et c'est un choix collectif largement respecté) et des normes de droit (normalisés par l'IETF)
  - Consignes pour les implémentations
- Autres organismes : IEEE (réseaux locaux)

Le modèle en couche

Définition

Le modèle en couche correspond à une manière de **décomposer un mécanisme de communication** en différents niveaux de détails indépendamment de l'implémentation



Le modèle OSI

Open Systems Interconnection

Le modèle OSI est une norme générale et abstraite (réf. ISO-7498) définissant un modèle en couche basé sur 7 couches distinctes

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data link
1. Physical

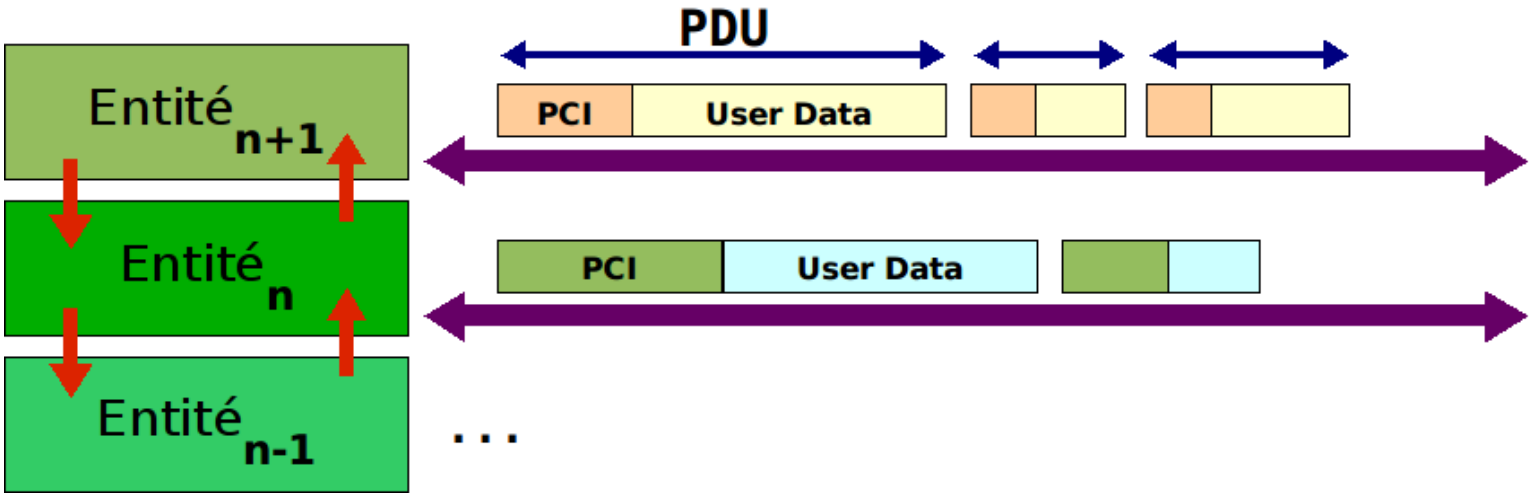
Vocabulaire du modèle OSI

PDU, PCI, UD

PDU (*Protocol Data Unit*)
Unité de données spécifique à un protocole de communication. Le `PDU(n)` désigne le PDU caractéristique de la couche `n`

PCI (*Protocol Control Information*)

UD (*User Data*)
Et on a : `PDU = PCI + UD`



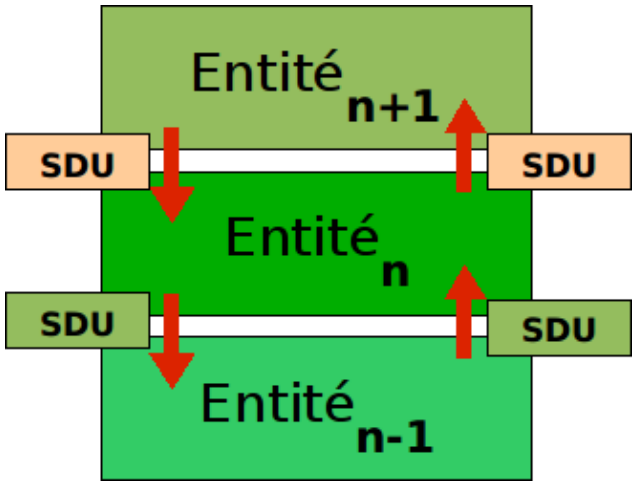
Vocabulaire du modèle OSI

SDU, fonctions, services

fonctions$_n$ Ensemble des traitements accomplis par l'entité de la couche *n*

service$_n$ Ensemble des tâches que peut fournir la couche *n* à la couche au dessus : $\{fonctions_n \cup service_{n-1}\}$

SDU (*Service Data Unit*)
Unité de données spécifique à un service. Le SDU_n désigne le *SDU* caractéristique de la couche *n*



Lois du modèle en couches

Loi de transparence des protocoles

Un $protocole_n$ doit être indépendant de la SDU_n

$\longrightarrow$ Permet de modifier le service fourni par la couche n sans impact sur le protocole n

Loi de transparence des services

Un $protocole_n$ doit être indépendant du $service_{n-1}$ dont il se sert

$\longrightarrow$ Permet de changer le $protocole_{n-1}$ à partir du moment où il offre les mêmes services

Les couches basses

Le support

Fils électriques, fibres optiques, ondes radio, etc.

La couche 1 : physique

La SDU de la couche 1 est le **bit**. Sa fonction est de transmettre des bits en les encodant sous forme de signaux de données. Les informations de contrôle PCI peuvent être hors-bande ou non

La couche 2 : lien

Le PDU de la couche 2 est l'**octet** ou la **trame** (bloc structuré de bits). Les fonctions de cette couche sont :

- La gestion du support (adressage, attribution, etc.)
- Le contrôle de flux et la détection d'erreurs

Les couches intermédiaires

La couche 3 : réseau

- Le PDU de la couche 3 est appelé **paquet** ou **datagramme**
- La fonction principale de cette couche est l'acheminement des paquets à travers un réseau de commutateurs

La couche 4 : transport

Au sein d'un réseau, la couche 4 fournit le même genre de services que la couche 2 : contrôle de flux, détection d'erreurs, etc.

Les couches hautes

La couche 5 : session

La couche 5 est responsable des échanges et de la gestion des connexions (ouverture, fermeture, problèmes de déconnexion, persistence des données, etc.). Par exemple : `RPC`

- La notion de session est souvent gérée au niveau applicatif (par exemple les sessions `HTTP` au travers des *cookies*

La couche 6 : présentation

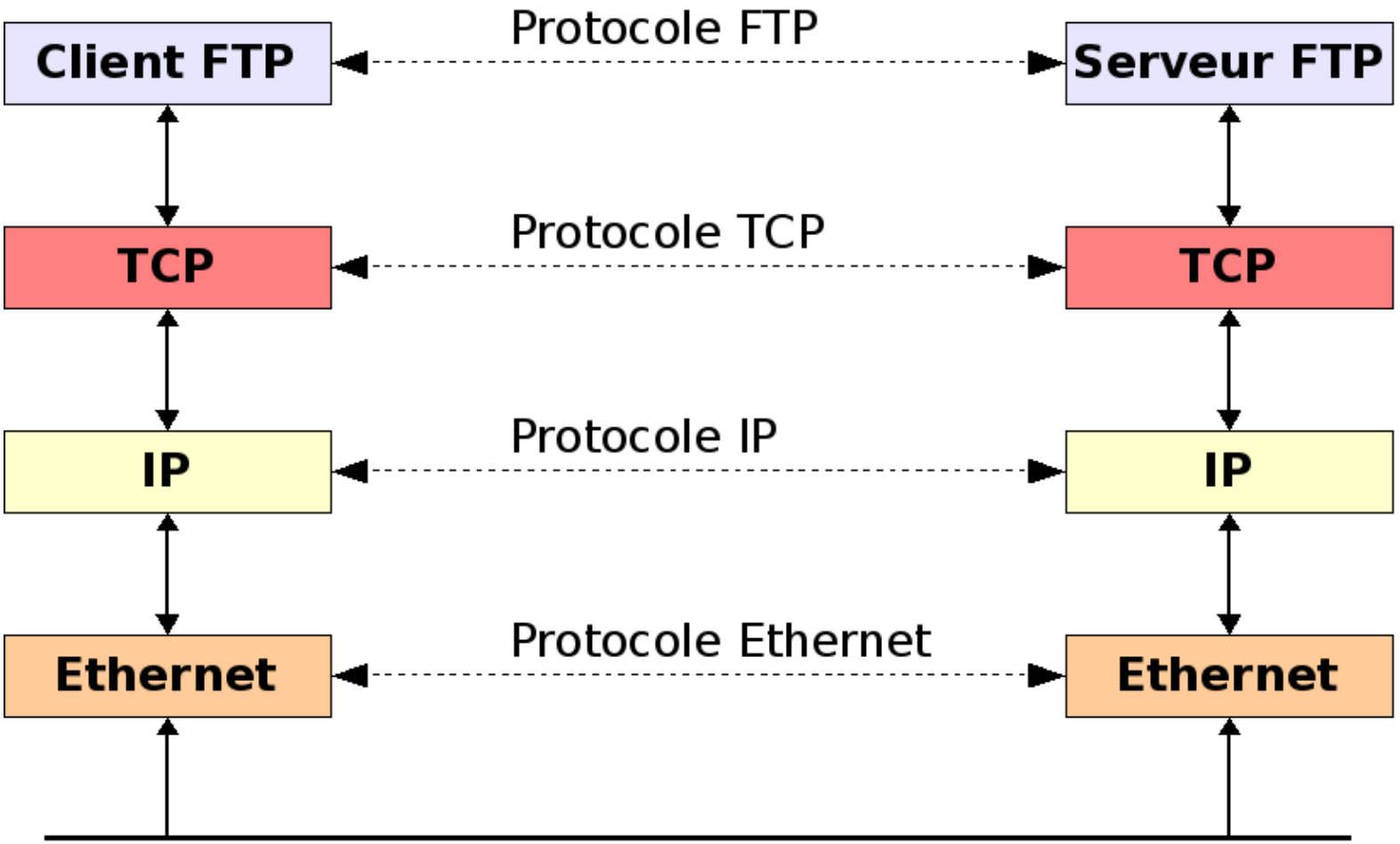
La couche 6 gère les problèmes de syntaxe ou de codage

- Par la conversion des données reçues de la couche supérieure, la couche présentation permet à deux applications de dialoguer même si elles utilisent des règles d'encodage différentes

La couche 7 : application

`SMTP`, `FTP`, `HTTP`, etc.

TCP/IP


Client FTP
Protocole FTP
Serveur FTP
TCP
Protocole TCP
TCP
IP
Protocole IP
IP
Ethernet
Protocole Ethernet
Ethernet

Le niveau 2

Couche de liaison de données

- Cette couche fournit une **liaison de données** entre des systèmes connectés au niveau physique
- Les principales fonctions fournies au niveau de la couche de lien sont le **partage du support** et la **délimitation des données**.
- La `PDU` peut être un octet (caractère) ou une trame (ensemble structuré d'octets)
- La couche de lien étant généralement très complexe, elle est souvent implémentée au moyen de deux sous-couches :

  MAC (*Media Access Control*)
  LLC (*Logical Link Control*)

Partage du support au niveau 2

- Différents types de liaisons :
  - Liaisons point à point
  - Liaisons multipoints (configurations en étoile)
  - Bus (plusieurs stations partagent le même support)
  - Configuration en anneau
- La complexité augmente en fonction du nombre de stations (fonction d'adressage)
- Si à un instant donné le protocole ne peut être utilisé que par une seule station, il faut :
  - Gérer un contrôle d'attribution pour éviter les collisions
  - Gérer les collisions dans le protocole

Délimitation des données

- La délimitation est indispensable au contrôle d'intégrité
- Repérage du début et de la fin d'une PDU
  - Des caractères spéciaux ajoutés au début et/ou à la fin d'un bloc peuvent jouer le rôle de marqueurs ou indiquer la longueur du bloc
- La délimitation peut aussi être héritées des couches 0 ou 1
- Relation temporelle entre l'emetteur et le récepteur
- Délimitation des champs à l'intérieur d'une PDU

Techniques d'encapsulation

Bourrage Rajout d'octets arbitraires quand le champ *user data* du `PDU(n)` est plus long que la `PDU(n+1)`

Segmentation Si le champ *user data* de le `PDU(n)` est plus court que la `PDU(n+1)`, le protocole segmente les données de la `PDU(n+1)` en plusieurs `PDU(n)`

Concaténation Le champ *user data* de le `PDU(n)` peut contenir plusieurs `PDU(n+1)`

Détection des erreurs

- Utilisation d'une somme de contrôle d'intégrité (*checksum*)
- Le *checksum* est calculé par l'émetteur et transmis avec la PDU. Cette somme est ensuite recalculée par le récepteur qui peut la comparer ainsi à la valeur transmise
- Une erreur de transmission peut néanmoins engendrer :
  - Un *checksum* erroné entraînant la retransmission
  - Une PDU jugée intègre à la réception
- Séquencements et acquittements

HDLC

High Level Data Link Component

HDLC est une norme (initialement ISO-3309) spécifiant différents services de la couche de lien :

1. La **délimitation des données** au moyen d'un drapeau inséré au début et à la fin de chaque trame (0x7E ou 01111110b)
2. La gestion de trames de différentes types : informations, supervision, etc.
3. Détection des erreurs de transmission au moyen d'un contrôle d'intégrité sur la trame (FCS)
4. Contrôle de flux (mécanisme d'acquittement)
5. Adressage

PPP

Point-to-Point Protocol

Protocole de niveau 2 spécifié par l'IETF (RFC 1661)

- Connexion directe ("point à point") entre 2 noeuds du réseau
- Largement déployé par les ISP : RTC, PPPoE, VPN, etc.
- Protocole en 4 sous-couches dont les 3 premières sont issues de HDLC (délimitation, intégrité, contrôle logique)
  - La couche supérieure de PPP gère le multiplexage des protocoles
  - Le protocole LCP (*Link Control Protocol*) permet d'établir, configurer et tester la connexion de lien de données
  - Le protocole NCP (*Network Control Protocol*) est spécifique à chaque couche réseau encapsulée

La couche Ethernet

Le terme *Ethernet* fait référence à un standard publié en 1982 (DEC, Intel et Xerox) puis normalisé par le comité 802 de l'IEEE quelques années plus tard (norme 802.3)

Bus à méthode d'accès CSMA/CD

- *Carrier Sense Multiple Access/Collision Detection*
- Chaque station peut emettre sur le bus. Une station détecte une collision quand ce qu'elle entend n'est pas identique à ce qu'elle a émis. Dans ce cas, la trame est réémise après un délai tiré au hasard dans un intervalle de plus en plus grand

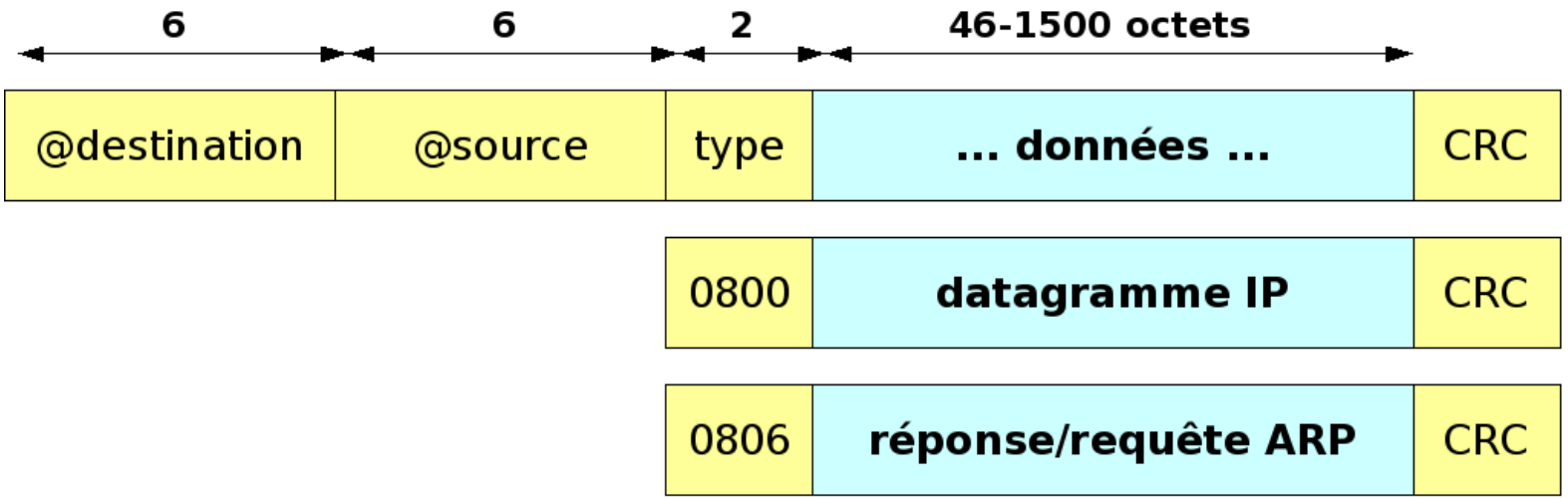
L'encapsulation des datagrammes IP est spécifiée :

- dans la RFC 894 pour les réseaux Ethernet
- dans la RFC 1042 pour les réseaux IEEE 802

Encapsulation Ethernet

- Les adresses ethernet sont codées sur 48 bits (6 octets)
- Les protocoles ARP et RARP effectuent la correspondance entre les adresses IP (32 bits) et les adresses matérielles
- Les trames ont une longueur minimale (octets de bourrage)
  - *Ethernet leak*
- Hubs et switchs

Format de la trame Ethernet



MTU

Maximum Transmission Unit

Le MTU est une caractéristique de la couche de lien spécifiant la taille maximale des données que l'on peut envoyer dans une trame

- Pour Ethernet, le MTU est 1500 (1492 pour 802.3)
- Pour les liaisons point à point type PPP, c'est une limite logique fixée en fonction de différents paramètres (temps de réponse souhaité, etc.)
- Au niveau réseau, on considère le MTU de chemin (plus petit MTU pour l'ensemble des chemins empruntés par les paquets réseaux dans un sens donné)

La couche 3 : réseau

La fonction principale de la couche 3 est l'acheminement des paquets (ou datagrammes) au travers d'un réseau

Caractéristiques

- Différents types d'éléments : routeurs et équipement terminaux
- Adressage des éléments au sein du réseau
- Commutation de circuits vs. commutation de paquets
- Mode connecté et mode non-connecté

Commutation de circuits et mode connecté

Une connexion réalisée par la couche 3 dans le mode commutation de circuit commence par la mise en place d'un **circuit virtuel**

- Les deux points du réseau sont **logiquement** connectés
- Cette connexion peut être permanente ou établie à la demande
- La couche 3 peut aussi prendre en charge le contrôle de flux et le séquencement

Commutation de paquets et mode non connecté

Dans ce mode de fonctionnement, les paquets sont acheminés indépendamment les uns des autres pour une connexion donnée

- Les informations nécessaires à l'acheminement des paquets sont contenues dans le paquet lui-même
- Pas de contrôle de flux, ni de séquencement

Commutation de paquets/circuits

- Obligation de moyens ou de résultats
- Différents niveaux de PCI (connexion / transmission)
- Routage
- Vitesse de commutation
- Contrôle de flux
- Séquencement / duplication
- Facturation

En raison de la qualité de service exigée par les nouvelles applications (`VoIP`, etc.), les deux modes sont utilisés
$\longrightarrow$ par exemple, au sein du coeur de réseau d'un opérateur, les datagrammes sont souvent encapsulés dans des circuits virtuels

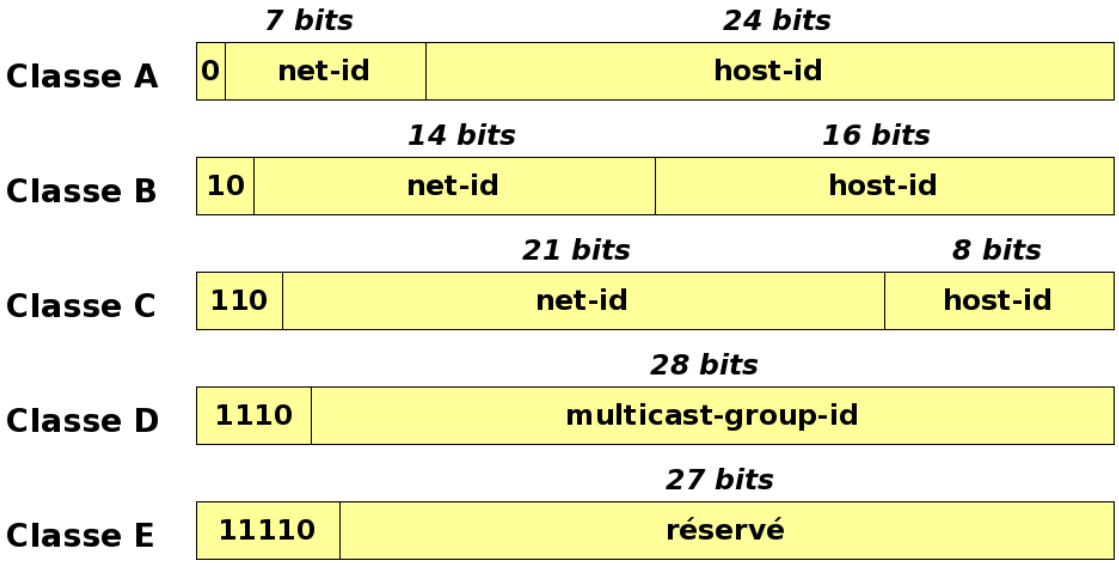
Internet Protocol (IP)

IP (défini dans la RFC 791) fournit un service de transport de datagrammes (commutation de paquets donc **sans connexion**) et **non fiable**

- Les données transportées peuvent être altérées
- Les datagrammes peuvent être perdus, dupliqués ou arriver dans le désordre
- La version 4 (IPV4) est la plus répandue

L'adressage IP

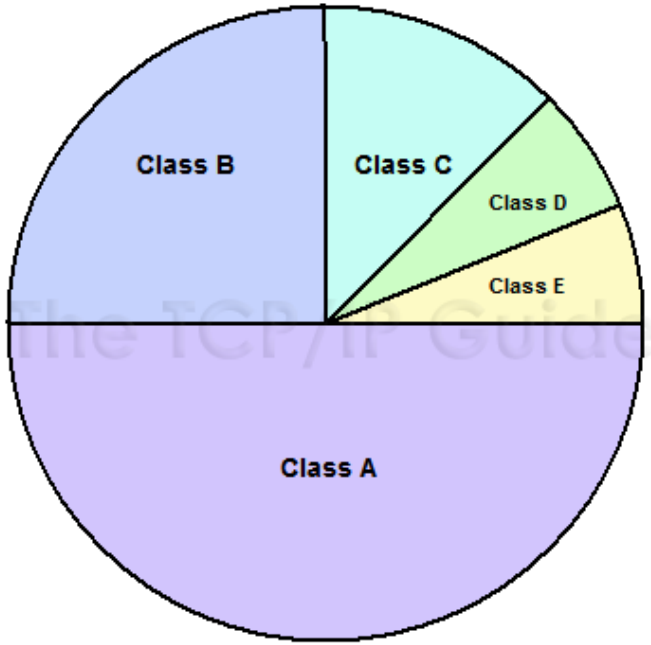
- Notation décimale pointée : W.X.Y.Z
- Une adresse peut être découpée en deux parties (identifiant réseau et identifiant machine), historiquement :



Les classes d'adresses IP

Découpage

- A : 0.0.0.0 à 127.255.255.255
- B : 128.0.0.0 à 191.255.255.255
- C : 192.0.0.0 à 223.255.255.255
- D : 224.0.0.0 à 239.255.255.255
- E : 240.0.0.0 à 247.255.255.255



Les adresses privées

- La RFC1918 spécifie plusieurs blocs d'adresses réservées aux réseaux privés (non routées sur Internet)

- 10.0.0.0-10.255.255.255 (10.0.0.0/8 - bloc 24 bits)
- 172.16.0.0-172.31.255.255 (172.16.0.0/12 - bloc 20 bits)
- 192.168.0.0-192.168.255.255 (192.168.0.0/16 - bloc 16 bits)

Broadcast et multicast

Broadcast et multicast sont mis en oeuvre lorsqu'une application souhaite envoyer un seul message à plusieurs récepteurs en même temps

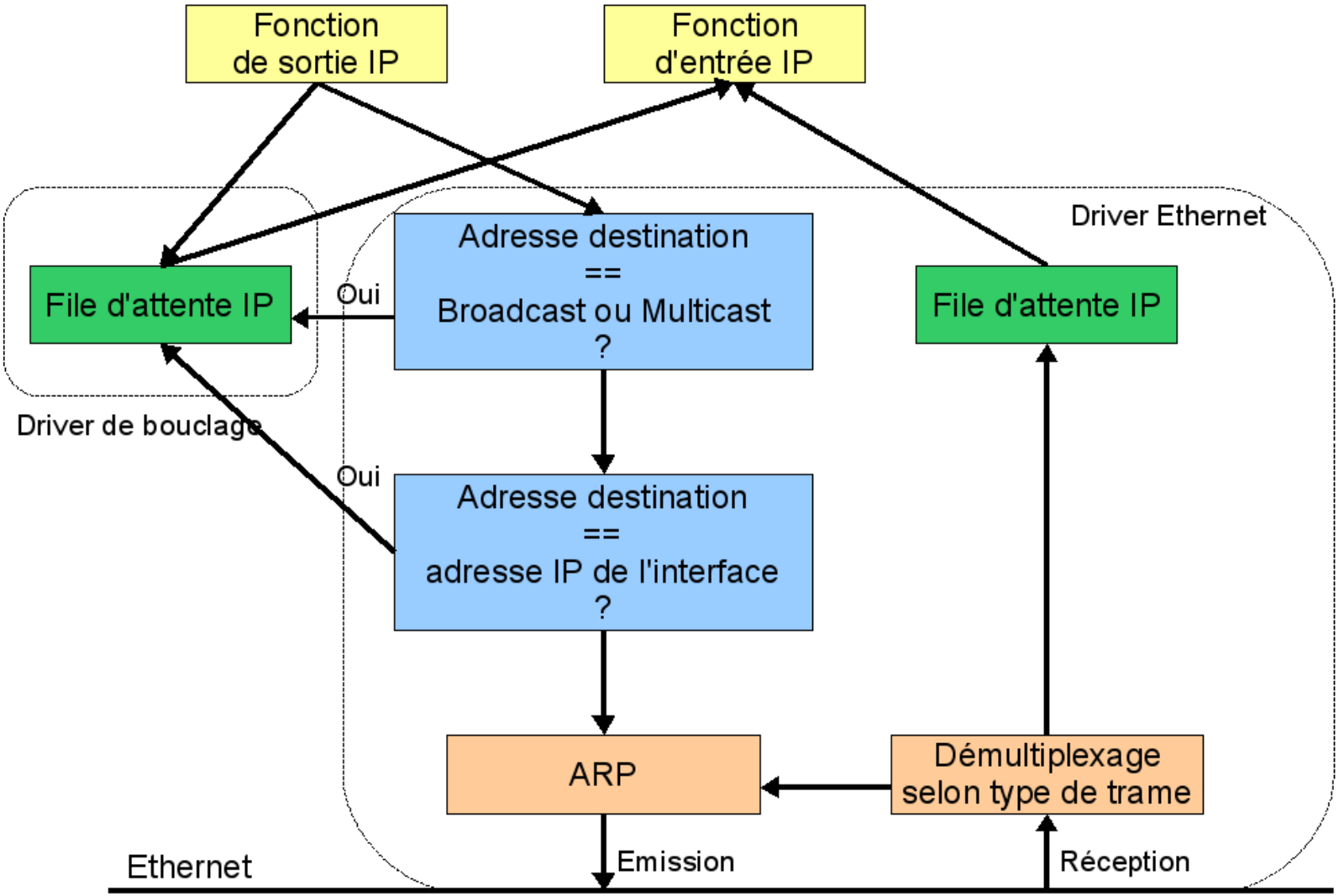
- Rappel : un filtrage est fait à tous les niveaux de la pile protocolaire
- Broadcast limité (255.255.255.255) et broadcast de réseau
- Multicast
  - Adresses de groupes : 224.0.0.0 à 239.255.255.255

Interface de bouclage

La plupart des implémentations supportent une interface de bouclage (*loopback interface*) permettant à un client et un serveur situés sur la même machine de communiquer via TCP/IP

- L'identificateur de réseau de classe A 127 est réservé au loopback
- Par convention, la plupart des systèmes utilisent l'adresse IP 127.0.0.1 et lui assignent le nom *localhost*
- Un datagramme envoyé à cette adresse IP ne doit apparaître dans aucun réseau

Traitement des datagrammes IP par le loopback



CIDR

Classless Inter-Domain Routing

CIDR est un nouveau découpage (proposé en 1993) de l'espace d'adressage IP afin de palier aux déficiences du découpage historique en classes A, B, etc.

- Utilisation de prefixes (masques de réseau) **de longueur variable**, permettant à la fois :
  - une gestion décentralisée (délégation)
  - une diminution de la taille des tables de routage (aggrégation)
- Format d'un bloc CIDR : A.B.C.D/n

CIDR

Mécanismes de délégation

- Les plus gros blocs (préfixes courts ou /8) sont directement gérés par l'IANA (*Internet Assigned Numbers Authority*)
- L'IANA délègue la gestion de chaque bloc aux RIRs (*Regional Internet Registries*) : RIPE, APNIC, ARIN, etc.
- Ces derniers délèguent ensuite les sous-blocs aux LIRs (*Local Internet Registries*) : ISP, universités, très grandes entreprises, etc.
- Généralement, les ISPs délèguent ensuite des blocs plus petits aux entreprises (de /24 à /29)

Découpage de sous-réseaux

Exercice n°1

Un hébergeur de services Internet loue un emplacement dans une baie ainsi que le bloc d'adresses `201.96.10.224/28`

- Quel est le nombre d'adresses IP utilisables ?
- Donner l'adresse de broadcast et le masque de sous-réseau

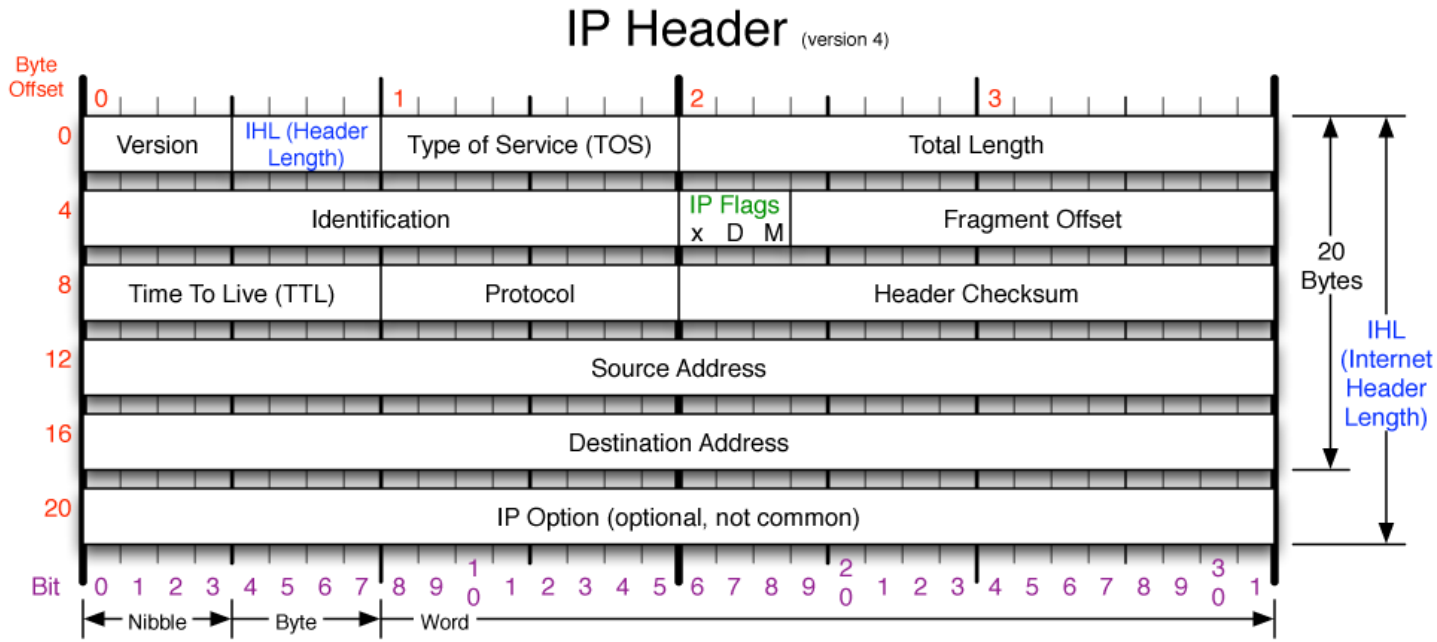
Exercice n°2

Découper le réseau `10.0.8.0/24` en trois sous-réseaux et donner pour chacun d'entre-eux l'adresse de réseau, le broadcast et le masque

La commande ifconfig sous OpenBSD

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33224
        groups: lo
        inet 127.0.0.1 netmask 0xff000000
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:04:23:bd:44:d2
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet 192.168.10.254 netmask 0xffffff00 broadcast 192.168.10.255
em1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:04:23:bd:51:31
        media: Ethernet autoselect (none)
        status: no carrier
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        lladdr 00:14:22:0b:d5:e0
        media: Ethernet autoselect (10baseT half-duplex)
        status: active
pppoe0: flags=8851<UP,POINTOPOINT,RUNNING,SIMPLEX,MULTICAST> mtu 1492
        dev: bge0 state: session
        sid: 0x1a45 PADI retries: 0 PADR retries: 0 time: 03:15:29
        groups: pppoe egress
        inet 213.41.244.114 --> 0.0.0.1 netmask 0xffffffff
```

Datagramme IP



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

x D M

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

RFC 791

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Fragmentation IP

La fragmentation IP intervient lorsqu'un datagramme IP est plus grand que le MTU de l'interface par laquelle il doit être envoyé

- La fragmentation peut se produire soit sur l'émetteur soit sur un routeur intermédiaire
- Le réassemblage se fait au niveau de la destination par la couche IP
  - La fragmentation est transparente pour les couches supérieures (malgré une dégradation possible de performances)
- Chaque fragment devient un paquet IP indépendant

Fragmentation IP

Principes

- Le champ **identification** contient une valeur unique pour chaque fragment IP que l'émetteur transmet
- Le bit **more fragments** est positionné à 1 excepté pour le dernier fragment
- Le champ **fragment offset** contient l'offset de ce fragment depuis le début du datagramme original
- Le champ **total length** est modifié pour contenir la taille de chaque fragment

ARP

Address Resolution Protocol

Au sein d'un même réseau local, l'adresse matérielle (de niveau 2) est utilisée pour la communication entre deux machines

ARP est un protocole de niveau 2 (définit dans la RFC 826) fournissant une correspondance **dynamique** entre les adresses de la couche réseau et les adresses de la couche de lien

- Typiquement sur un LAN : trouver l'adresse Ethernet (48 bits) d'une station à partir de son adresse IP (32 bits)

ARP

Fonctionnement

On suppose que la station 192.168.1.10 souhaite envoyer un datagramme IP à la machine 192.168.1.254 ; elle doit pour cela connaître l'adresse matérielle de la destination

- Elle envoie une requête ARP à **toutes** les machines connectées au support (*broadcast*)
- La requête contient la question : "que celui (ou celle) qui a l'adresse IP 192.168.1.254 se dénonce immédiatement !"
- La machine 192.168.1.254 reçoit la requête, et envoie une réponse (destinée uniquement à l'emetteur) contenant son adresse IP et son adresse matérielle

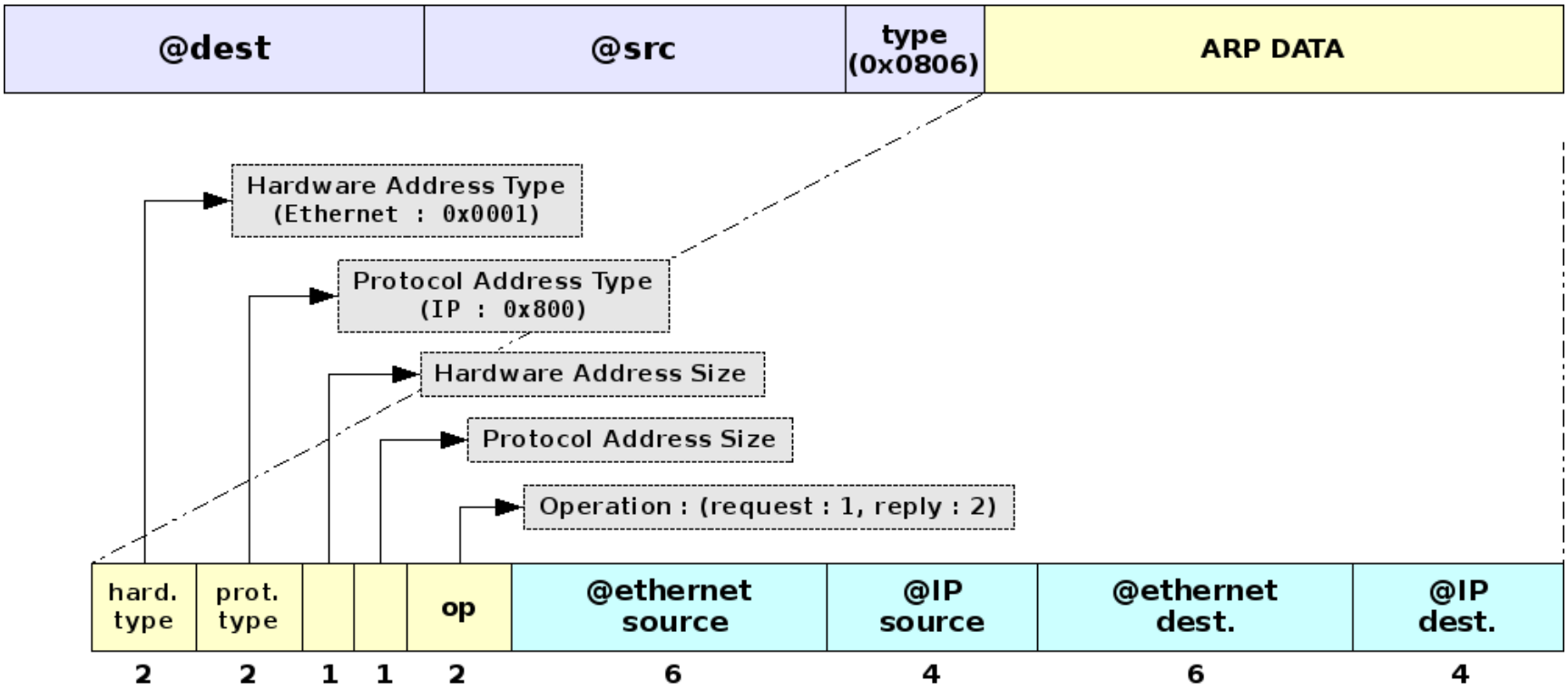
ARP

Remarques

- ARP est un protocole général et fonctionne avec tout type de réseau supportant le broadcast : Token Ring, FDDI, etc.
- Pour des raisons d'efficacité, chaque station dispose d'un cache contenant les correspondances récemment utilisées
  - Le cache de la station destination est mis à jour lorsqu'elle reçoit la requête (`@ADR_IP_SRC -> @ADR_MAC_SRC`)
  - Le cache de la machine source est mis à jour lorsqu'elle reçoit la réponse (`@ADR_IP_DST -> @ADR_MAC_DST`)
- ARP Gratuit (*Gratuitous ARP*)
- Le protocole **Reverse ARP** ou RARP permet l'association inverse, i.e. associer une adresse réseau à une adresse de niveau 2

ARP

Format de la trame ARP



Travaux dirigés

Décodage des paquets ARP

```
0x0000:  ffff ffff ffff 0002 448c 3d50 0806 0001
0x0010:  0800 0604 0001 0002 448c 3d50 58ab c84a
0x0020:  0000 0000 0000 58ab c8fe 0000 0000 0000
0x0030:  0000 0000 0000 0000 0000 0000
```

```
0x0000:  0002 448c 3d50 0007 cb28 6c0a 0806 0001
0x0010:  0800 0604 0002 0007 cb28 6c0a 58ab c8fe
0x0020:  0002 448c 3d50 58ab c84a 0000 0000 0000
0x0030:  0000 0000 0000 0000 0000 0000
```

ARP pour la redondance réseau

La mise en place d'équipements redondants permet d'améliorer la **disponibilité** d'un système, que ce soit au niveau de la reprise sur panne (*fail-over*) ou bien de l'équilibrage de charge (*load-balancing*)

- Exemples de protocoles :

  VRRP (*Virtual Router Redundancy Protocol*)
  HSRP (*Hot Swap Redundancy Protocol*)
  CARP (*Common ARP Redundancy Protocol*)
- La redondance au niveau réseau est une notion différente de la redondance applicative
  - Pas de gestion des états (*stateless*)
  - Peu d'intelligence donc peu de complexité

Routage IP

Principe

Si la machine connaît la destination (connexion directe), elle envoie directement le datagramme, sinon elle l'envoie à une autre machine (routeur) et le laisse se débrouiller.

- Une station ne réémet jamais un datagramme
- IP ne connaît la route complète d'aucune destination non connectée directement à la machine

Routage IP dans la pile OSI

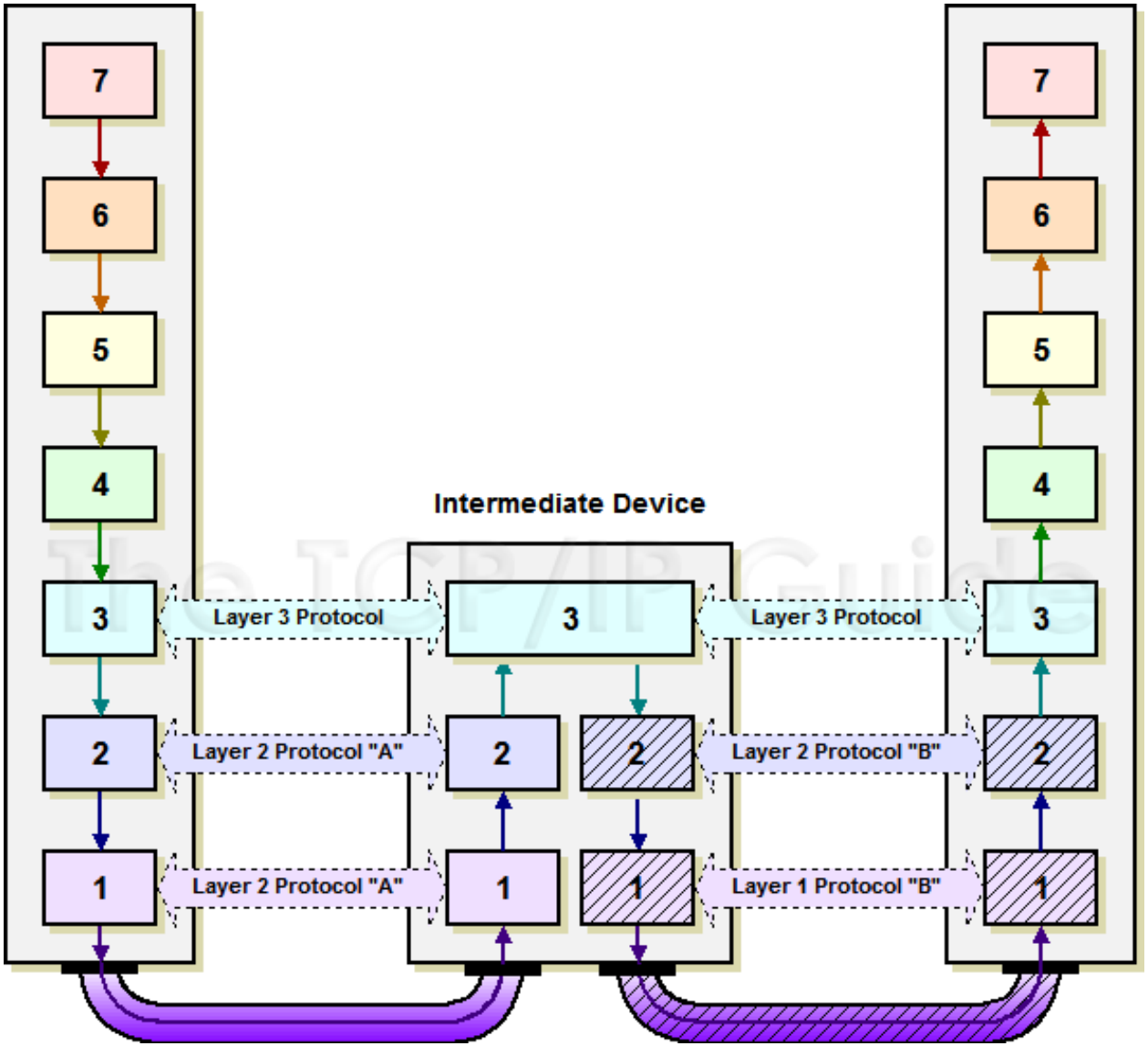


Table de routage

- Chaque ligne de la table de routage contient les informations suivantes :
  - Destination (adresse de station ou de réseau)
  - Routeur de saut suivant (*next hop router*)
  - Spécification de l'interface de sortie
- Importance de la qualité de la table de routage

Exemple de table de routage

Destination	Gateway	Flags	Refs	Use	Mtu	Prio	Iface
default	88.171.200.254	UGS	5	1067787	-	8	rl0
10.0.20.0/25	10.0.20.193	UGS	4	2214988	-	8	vr0
10.0.20.128/26	10.0.20.193	UGS	0	0	-	8	vr0
10.0.20.192/26	link#2	UC	1	0	-	4	vr0
10.0.20.193	00:15:f2:3d:7c:55	UHLc	4	5056	-	4	vr0
88.171.200/24	link#1	UC	1	0	-	4	rl0
88.171.200.74	127.0.0.1	UGHS	0	0	33204	8	lo0
88.171.200.254	00:07:cb:28:6c:0a	UHLc	2	1577	-	4	rl0
127/8	127.0.0.1	UGRS	0	0	33204	8	lo0
127.0.0.1	127.0.0.1	UH	4	73759	33204	4	lo0
224/4	127.0.0.1	URS	0	0	33204	8	lo0

Routage fixe

Routage fixe

- La table de routage est fixe (réseau local)
- Support mal la panne ou la congestion

Routage adaptatif

Routage adaptatif centralisé

- Un superviseur centralise l'information sur l'état des commutateurs et des lignes, il décide du routage et informe les équipements en temps-réel
- La centralisation évolue vers une hiérarchie pour les gros réseaux
- Les informations transitent par le réseau supervisé lui-même

Routage adaptatif décentralisé

- Chaque noeud maintient une table d'état des liens avec ses voisins
- Méthodes complexes, ce routage peut entraîner des boucles (RIP)

Le routage dans Internet

En quelques mots...

Le réseau est structuré en un ensemble de systèmes autonomes appelés `AS`. Chaque AS peut être défini comme un ensemble de réseaux `IP` (préfixes `CIDR` distincts) gérés par une même entité.

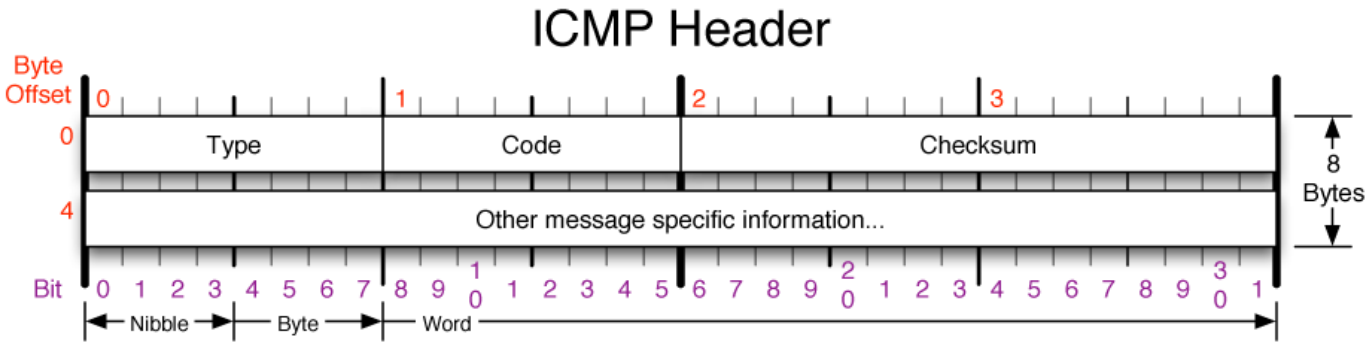
- Les routeurs situés aux frontières de chaque AS échangent avec leurs voisins des politiques de routage en s'appuyant sur le protocole `BGP` (*Border Gateway Protocol*)
- A l'intérieur de chaque AS, les protocoles de routage classiques sont utilisés : `RIP`, `OSPF`, etc.

ICMP

Internet Control Message Protocol

- ICMP communique les messages d'erreur et d'informations au niveau IP (spécifié dans la RFC 792)
- Un paquet ICMP peut encapsuler une requête ou un message d'erreur et on lui associe un type et un code, ainsi qu'un contenu de (dépendant de ce type et de ce code)
- Une erreur ICMP contient toujours l'entête IP et les 8 premiers octets du datagramme qui a causé l'erreur

Format du paquet ICMP



ICMP Message Types

Type	Code/Name
0	Echo Reply
3	Destination Unreachable
	0 Net Unreachable
	1 Host Unreachable
	2 Protocol Unreachable
	3 Port Unreachable
	4 Fragmentation required, and DF set
	5 Source Route Failed
	6 Destination Network Unknown
	7 Destination Host Unknown
	8 Source Host Isolated
	9 Network Administratively Prohibited
	10 Host Administratively Prohibited
	11 Network Unreachable for TOS
	12 Host Unreachable for TOS
	13 Communication Administratively Prohibited
4	Source Quench
5	Redirect
	0 Redirect Datagram for the Network
	1 Redirect Datagram for the Host
	2 Redirect Datagram for the TOS & Network
	3 Redirect Datagram for the TOS & Host
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
	0 TTL Exceeded in Transit
	1 Fragment Reassembly Time Exceeded
12	Parameter Problem
	0 Pointer indicates the error
	1 Missing a Required Option
	2 Bad Length
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
30	Traceroute

Checksum

Checksum of entire UDP segment and pseudo header (parts of IP header) (for UDP)

Checksum of ICMP header (for ICMP)

RFC 768 and 792

Please refer to RFC 768 for the complete User Datagram Protocol (UDP) Specification, and to RFC 792 for the Internet Control Message protocol (ICMP) specification.

Le programme `ping`

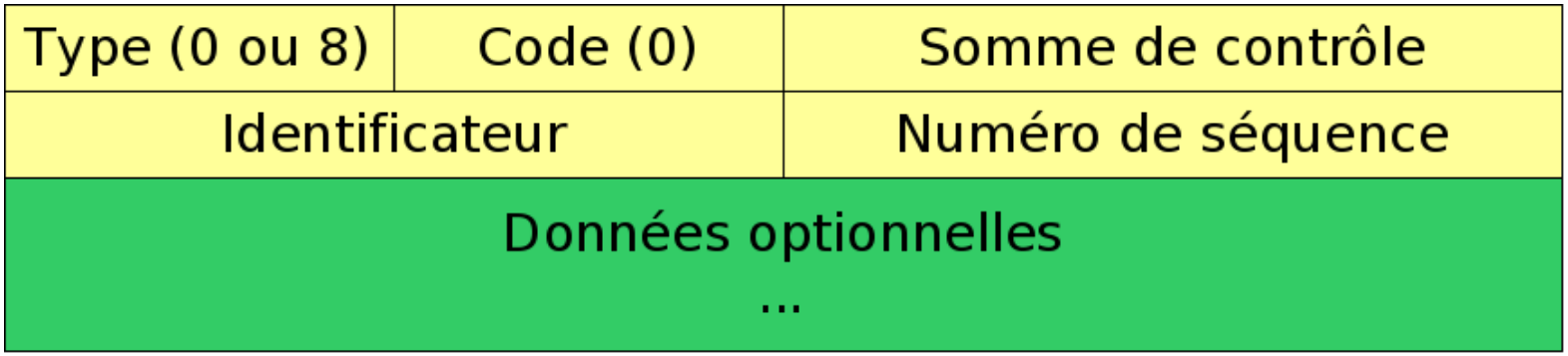
Le programme `ping` est un outil de diagnostic permettant de vérifier si une machine est accessible

- Envoi d'une requête ICMP de type **echo-request**
- Attente d'une réponse ICMP de type **echo-reply**

Remarque

Les requêtes ICMP sont souvent filtrées pour des raisons de sécurité, donc si une machine ne répond pas aux requêtes ping, cela ne signifie pas que les applications réseaux s'exécutant sur cette machine ne seront pas accessibles

Les requêtes ICMP `echo-request` et `echo-reply`



- L'association requêtes/réponses : identifiant et num. séquence
- Calcul du RTT (*round-trip time*) par enregistrement de la date d'envoi dans les données optionnelles
- L'OS peut parfois être identifié en étudiant les données optionnelles

Travaux dirigés

Décodage de paquets ICMP

```
0x0000:  0007 cb28 6c0a 0002 448c 3d50 0800 4500
0x0010:  0054 9571 0000 ff01 738c 58ab c84a d5fb
0x0020:  bbb9 0800 905e 8620 0000 4754 0d62 000c
0x0030:  a1bb 0809 0a0b 0c0d 0e0f 1011 1213 1415
0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425
0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435
0x0060:  3637
```

```
0x0000:  0002 448c 3d50 0007 cb28 6c0a 0800 4500
0x0010:  0054 db3c 0000 3301 f9c1 d5fb bbb9 58ab
0x0020:  c84a 0000 985e 8620 0000 4754 0d62 000c
0x0030:  a1bb 0809 0a0b 0c0d 0e0f 1011 1213 1415
0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425
0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435
0x0060:  3637
```

Le programme traceroute

Le programme traceroute remplace l'option IP "*Record Route*" devenue obsolète et fournit le même service en utilisant le champ TTL de l'entête et les messages ICMP

Lorsqu'un routeur reçoit un datagramme IP dont le champ TTL est à 0 ou 1, il ne doit pas le retransmettre. Au lieu de cela, le routeur élimine le datagramme et envoie à l'émetteur un message ICMP de type time-exceeded

Exemple de résultats de traceroute

```
[pascal@bean:~]$ traceroute -n www.google.com
traceroute to www.l.google.com (64.233.183.104)
 1  88.171.200.254 36.758 ms  37.214 ms  37.177 ms
 2  78.254.0.158   36.754 ms  36.278 ms  37.55 ms
 3  78.254.255.17  36.923 ms  36.564 ms  36.837 ms
 4  78.254.255.13  36.926 ms  37.44 ms   40.297 ms
 5  78.254.255.9   37.199 ms  36.813 ms  36.600 ms
 6  78.254.255.5   37.147 ms  36.832 ms  36.793 ms
 7  * * *
 8  212.27.50.173  38.36 ms   *          39.800 ms
 9  212.27.51.10   37.341 ms  37.264 ms  37.326 ms
10  213.228.3.136  80.496 ms  69.623 ms  53.636 ms
11  72.14.232.104  47.984 ms  47.647 ms  47.184 ms
12  72.14.238.119  55.143 ms  55.531 ms  55.530 ms
13  64.233.175.246 56.891 ms  55.790 ms  56.36 ms
14  72.14.233.81   57.412 ms  56.512 ms  56.598 ms
15  216.239.43.30  69.33 ms   62.444 ms  63.599 ms
16  64.233.183.104 56.132 ms  56.773 ms  55.790 ms
```

Découverte de la topologie d'un réseau

Dans le cadre d'un audit de sécurité orienté réseau, la découverte de la topologie du réseau IP est généralement une des premières étapes effectuée par les consultants en sécurité

- Les outils de base (`ping`, `traceroute`, `tcpdump`, etc.) permettent d'avoir un premier aperçu de la structure du réseau
- D'autres outils plus spécialisés peuvent ensuite être utilisés
  - `scapy`, `nmap`, `hping`, etc.
- La découverte des couches supérieures (scans applicatifs) fournit également des informations intéressantes

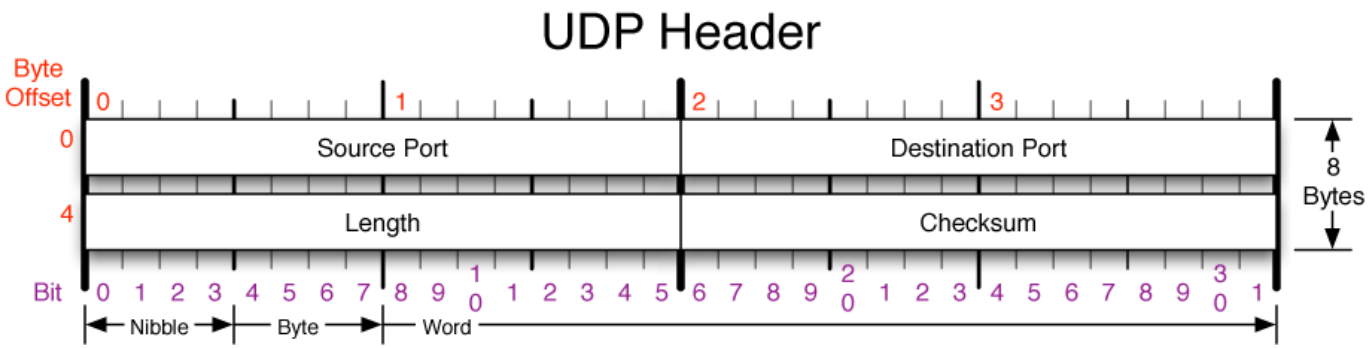
UDP

User Datagram Protocol

UDP est un protocole de couche de transport simple, orienté datagramme

- Exactement un datagramme est généré pour chaque opération de sortie effectuée par un processus (contrairement à un protocole orienté flux)
- Aucune garantie de fiabilité
- La taille des datagrammes émis est de la responsabilité de l'application
- La somme de contrôle inclut l'entête UDP et les données

Format du datagramme



TCP

Transmission Control Protocol

Les spécifications originales sont données par la RFC 793 (1981) mais d'autres spécifications apportent des informations complémentaires et des extensions.

TCP est un protocole **orienté connexion**, **fiable**, et fournissant un **flux d'octets** aux couches supérieures

end-to-end argument

Le design de TCP est principalement basé sur l'idée que l'intelligence du système doit reposer sur les deux extrémités et non sur les couches de liens et les équipements intermédiaires

Flux d'octets

TCP est un service à flux d'octets (*byte stream service*) : si l'application écrit 30 octets en 3 opérations de 10 octets, l'application destinataire ne peut pas connaître la taille des écritures individuelles

Orienté connexion

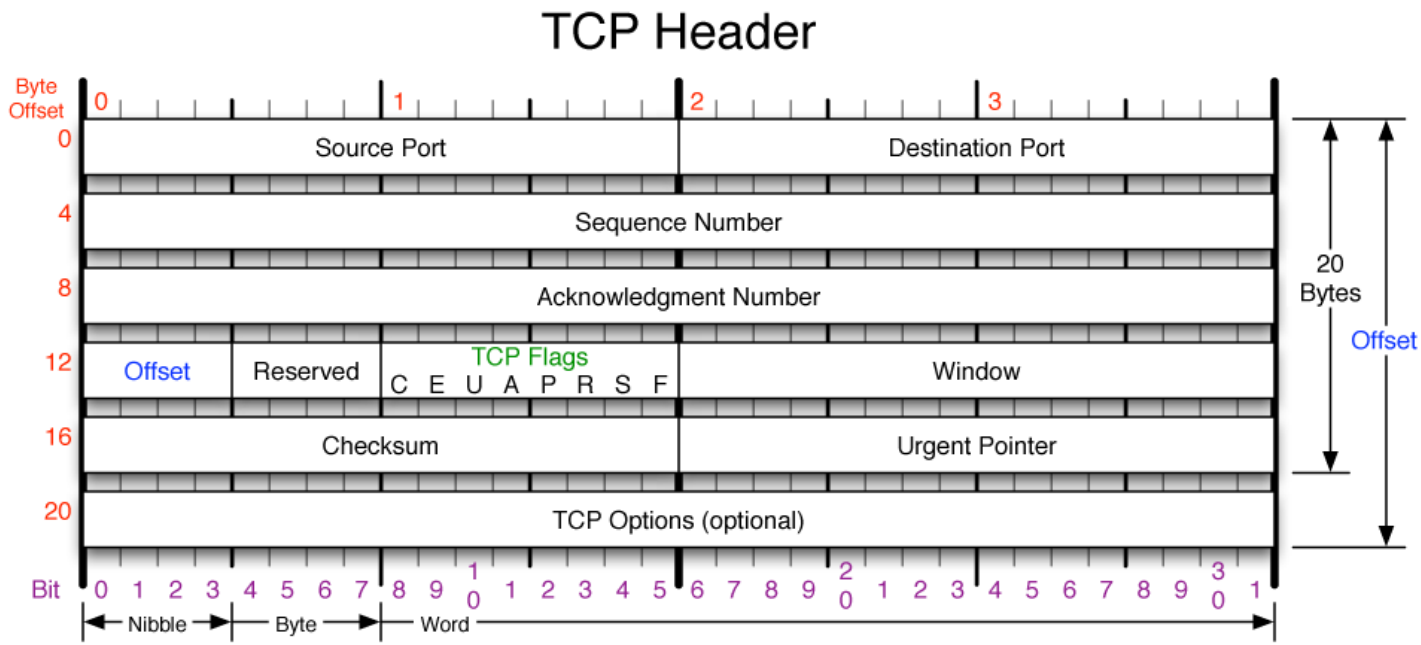
Avant que des données puissent être échangées en TCP, les deux extrémités doivent se mettre d'accord (se synchroniser)

TCP

Principes généraux

- Les données applicatives sont fractionnées par TCP en fragments de taille adéquate (l'unité de base est le **segment**)
- Après émission d'un segment, TCP déclenche un *timer* de retransmission et attend un acquittement
- TCP maintient une somme de contrôle de son entête et de ses données
- Les fragments TCP sont réordonnés à l'arrivée
- TCP gère également le contrôle de flux et les paquets dupliqués

L'entête TCP



TCP Flags

C E U A P R S F

Congestion Window
C 0x80 Reduced (CWR)
E 0x40 ECN Echo (ECE)
U 0x20 Urgent
A 0x10 Ack
P 0x08 Push
R 0x04 Reset
S 0x02 Syn
F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Reciever Response	1 1	0 1
Sender Response	1 1	1 1

TCP Options

0 End of Options List
1 No Operation (NOP, Pad)
2 Maximum segment size
3 Window Scale
4 Selective ACK ok
8 Timestamp

Checksum

Checksum of entire TCP segment and pseudo header (parts of IP header)

Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

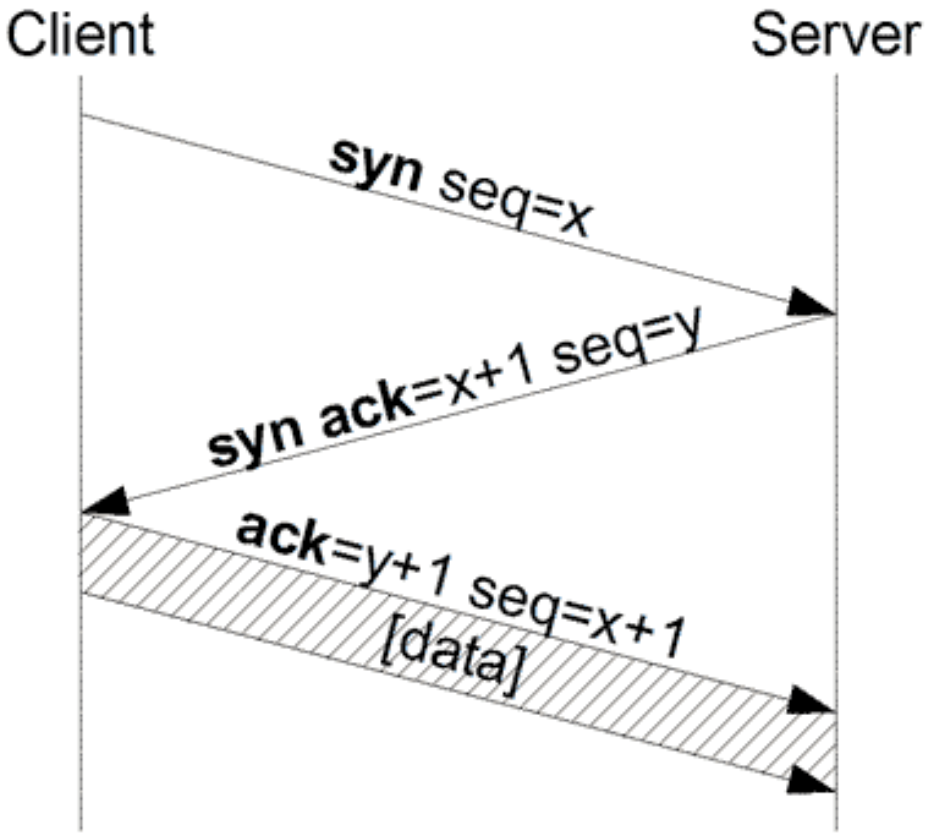
Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.

Établissement d'une connexion TCP

Poignée de main en trois étapes

- Le client émet un segment SYN spécifiant le numéro de port du serveur vers lequel il veut se connecter ainsi que le numéro de séquence initial (NSI)
- Le serveur répond avec son propre segment SYN contenant également son numéro de séquence initial. Dans le même segment, le serveur acquitte le SYN du client : ACK (NSI(client) + 1)
- Le client acquitte le SYN du serveur : ACK (NSI(serveur) + 1)

TCP : *three way handshake*



Fermeture d'une connexion TCP

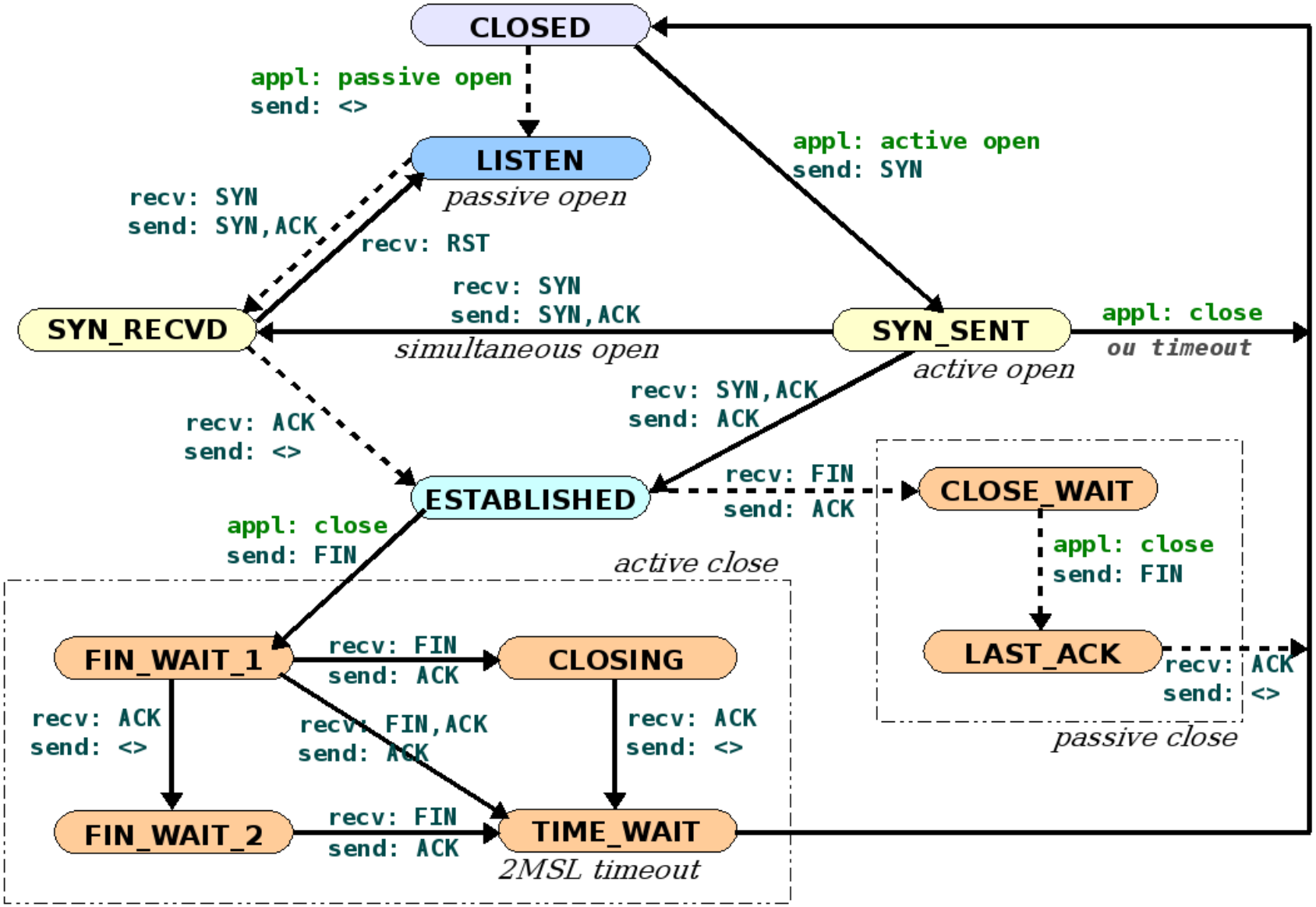
Il faut 4 segments pour fermer une connexion TCP

Rappel

TCP est un protocole *full-duplex* : les données peuvent s'écouler dans les deux sens de façon indépendant

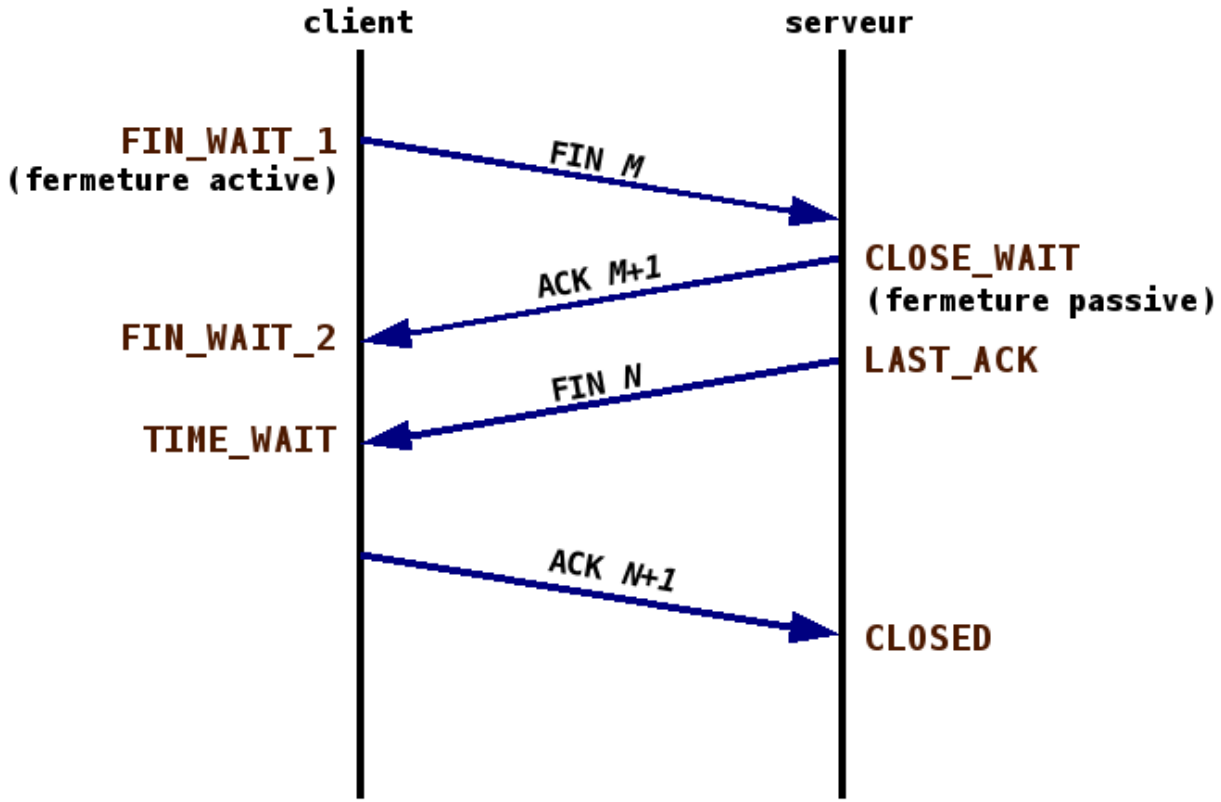
- N'importe quelle extrémité peut envoyer un FIN lorsqu'elle est en train d'émettre des données. A la réception du FIN, TCP notifie à l'application au dessus que l'autre extrémité a fini d'envoyer des données
- La réception du FIN signifie **seulement** qu'il n'y a plus de données dans cette direction
- Notions de fermeture active et fermeture passive

Diagramme d'états TCP



TCP

Terminaison d'une connexion



L'état `TIME_WAIT` est aussi appelé **état d'attente 2MSL**

La durée de vie maximale pendant laquelle un segment peut exister sur le réseau est appelée MSL (*Maximum Segment Lifetime*)

- Limitation supplémentaire par rapport au TTL de la couche IP
- Valeurs inférieures à 2mns pour la plupart des implémentation

Lorsque qu'il ne reste plus de données à envoyer (après une fermeture active) et après l'envoi du `ACK` final, TCP doit entrer dans l'état `TIME_WAIT` et y rester pendant **2 x MSL**

- Si le `ACK` final est perdu, il pourra alors être réémis
- Durant ce laps de temps, la socket ne peut pas être utilisée

Algorithme de Nagle

L'algorithme de Nagle (proposé dans la RFC 896) propose une solution simple pour optimiser le trafic réseau résultant de l'émission de paquets de petites tailles (*tinygrams*).

Une connexion TCP ne peut avoir qu'un seul petit segment non acquitté

- On considère un segment comme "petit" si sa longueur est inférieure au MSS
- Cet algorithme peut être dévalidé avec l'option `TCP_NODELAY`

Problèmes de sécurité inhérents de TCP/IP

La couche TCP/IP est vulnérable à différentes menaces pouvant porter atteinte à :

- l'**intégrité** et à la **confidentialité** des données
- l'**imputabilité** des acteurs
- la **disponibilité** des systèmes

Les protocoles basés sur TCP/IP (i.e. les couches supérieures) doivent prendre en compte et traiter ces menaces si ces dernières représentent un risque

Interception du trafic réseau

La couche TCP/IP fournit un moyen de transport sur le réseau et les données transitent par un certain nombre de noeuds

$\longrightarrow$ un attaquant **ayant le contrôle** d'un des noeuds de ce chemin peut mener tous les types d'attaques possibles

- La plupart des attaques ont pour premier objectif de **détourner le trafic** et de le faire passer par un point du réseau contrôlé par l'attaquant
- Par exemple : `ARP` *spoofing*, `DHCP`, etc.

ARP spoofing

ARP cache poisoning, ARP poison routing

- ARP est un protocole sans état car il n'y a pas de notion de transaction entre les requêtes et les réponses
- Une réponse ARP reçue par une station entraîne une mise à jour de son cache

→ En envoyant des fausses réponses ARP, un attaquant peut forcer la mise à jour des caches des stations

Injection de trafic réseau

On se place maintenant dans le cas où l'attaquant ne peut pas intercepter directement le trafic entre deux points du réseau

L'objectif de l'attaquant est de se faire passer pour une des deux machines et d'envoyer des données TCP valides (i.e. qui seront acceptées par le destinataire)

Pour cela, il doit être capable :

- d'envoyer les paquets sur le réseau en prenant comme adresse IP source l'adresse d'un des deux protagonistes
- de deviner le **numéro de séquence** correct pour la trame que l'on veut envoyer
- de faire en sorte que le destinataire reçoive en premier la fausse trame TCP

Les attaques sur la disponibilité

Les attaques de type `SYN` floods permettent de paralyser la couche TCP/IP de la victime

- Saturation des files d'attente de la couche TCP pour les connexions à moitié établies (état `SYN_RECVD`)
- Attaque plus efficace avec des adresses sources forgées (`IP` spoofing)

Les attaques de type *sockstress* ont pour conséquence un épuisement des ressources du serveur

TCP et la confidentialité des données

TCP ne fournit aucune garantie concernant la confidentialité des données (mais ce n'est pas vraiment le rôle d'une couche de transport)
Solutions possibles :

- Chiffrement au niveau IP avec IPSEC
- Chiffrement au niveau applicatif avec SSL