

Programmation Réseaux

Année 2022-2023

Anne Micheli

Evaluations :

- partiel le vendredi 10 mars
- projet donné vers la 5ème ou 6ème semaine de cours
- examen

Evaluations :

- partiel le vendredi 10 mars
- projet donné vers la 5ème ou 6ème semaine de cours
- examen

Les travaux en fin de semestre :

- TP des trois dernières semaine délocalisés
- ⇒ configuration de la connexion ssh aux machines de l'UFR depuis votre ordinateur personnel impérative !

Tous les TP commencent la semaine prochaine :

- Info 1 - mardi 16h15 - Pierre Charbit
- Info 2 - lundi 14h - Aldric Degorre
- Info 3 - lundi 10h45 - Yaelle Vincont
- Info 4 - mercredi 8h30 - Florian Renkin
- Info 5 - jeudi 16h15 - Colin Gonzalez
- MI - vendredi 16h30 - Anne Micheli

Organisation

Tous les TP commencent la semaine prochaine :

- Info 1 - mardi 16h15 - Pierre Charbit
- Info 2 - lundi 14h - Aldric Degorre
- Info 3 - lundi 10h45 - Yaelle Vincont
- Info 4 - mercredi 8h30 - Florian Renkin
- Info 5 - jeudi 16h15 - Colin Gonzalez
- MI - vendredi 16h30 - Anne Micheli

Si ce n'est déjà fait, inscrivez-vous au cours sur **Moodle**. Vous y trouverez documentation de cours, énoncés de TP et projet, annonces... Nous pourrons utiliser Moodle également pour vous écrire
⇒ **inscrivez-vous** sur Moodle dans votre **groupe de TP**.

Objectifs du cours

C'est un cours de programmation réseau et non de réseau.

- Comprendre les mécanismes de la communication « haut-niveau » entre des machines
- Faire communiquer des machines
- Formater des données
- Programmer en C

I- Introduction au réseau

Réseau

Un réseau est formé d'entités (ordinateur, routeur, LAN) reliées entre elles par câble, fibre, wifi, ...

Réseau

Un réseau est formé d'entités (ordinateur, routeur, LAN) reliées entre elles par câble, fibre, wifi, ...

Routeur

Un routeur permet de transmettre des paquets d'une interface à l'autre.

Réseau

Un réseau est formé d'entités (ordinateur, routeur, LAN) reliées entre elles par câble, fibre, wifi, ...

Routeur

Un routeur permet de transmettre des paquets d'une interface à l'autre.

Interface

Une interface réseau assure la connexion entre une machine et un réseau donné.

Réseau

Réseau

Un réseau est formé d'entités (ordinateur, routeur, LAN) reliées entre elles par câble, fibre, wifi, ...

Routeur

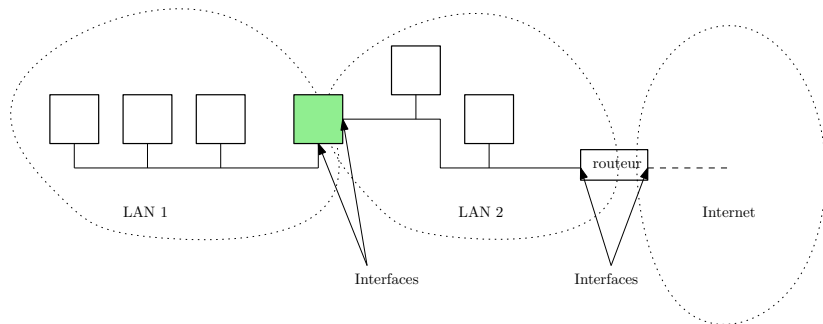
Un routeur permet de transmettre des paquets d'une interface à l'autre.

Interface

Une interface réseau assure la connexion entre une machine et un réseau donné.

LAN

Local Area Network : réseau local



Il y a deux types d'adresse :

Il y a deux types d'adresse :

- adresse MAC (Media Access Control) : adresse physique d'une entité stockée en général dans la carte réseau. Utilisée pour l'adressage sur un même réseau local. L'adresse est unique et pérenne.

Il y a deux types d'adresse :

- adresse MAC (Media Access Control) : adresse physique d'une entité stockée en général dans la carte réseau. Utilisée pour l'adressage sur un même réseau local. L'adresse est unique et pérenne.
- adresse IP (Internet Protocol) : adresse donnée temporairement à une entité sur un réseau donné. Une entité peut avoir plusieurs adresses IP éventuellement sur des réseaux différents.

Adresse IPv4

L'adresse est codée sur 32 bits.

L'adresse est codée sur 32 bits.

Représentation : 4 entiers non signés codés chacun sur 8 bits donc chaque entier a une valeur entre 0 et 255. Les entiers sont séparés par un point.

L'adresse est codée sur 32 bits.

Représentation : 4 entiers non signés codés chacun sur 8 bits donc chaque entier a une valeur entre 0 et 255. Les entiers sont séparés par un point.

Exemple : 10.0.175.225

L'adresse est codée sur 32 bits.

Représentation : 4 entiers non signés codés chacun sur 8 bits donc chaque entier a une valeur entre 0 et 255. Les entiers sont séparés par un point.

Exemple : 10.0.175.225

L'adresse **loopback** : 127.0.0.1

adresse locale de l'entité. Les paquets ne sont pas envoyés sur le réseau mais vers une application locale.

L'adresse est codée sur 32 bits.

Représentation : 4 entiers non signés codés chacun sur 8 bits donc chaque entier a une valeur entre 0 et 255. Les entiers sont séparés par un point.

Exemple : 10.0.175.225

L'adresse **loopback** : 127.0.0.1

adresse locale de l'entité. Les paquets ne sont pas envoyés sur le réseau mais vers une application locale.

$2^{32} = 4294967296$ adresses IP distinctes \Rightarrow pas assez d'adresses \Rightarrow nécessité d'utiliser un autre ensemble d'adresses

L'adresse est codée sur 128 bits.

L'adresse est codée sur 128 bits.

Représentation : 8 entiers hexadécimales codés chacun sur 16 bits. Les entiers sont séparés par « : ».

Exemple : 1b01:fa06:0842:1510:fc:a41b:189a:31af

L'adresse est codée sur 128 bits.

Représentation : 8 entiers hexadécimaux codés chacun sur 16 bits. Les entiers sont séparés par « : ».

Exemple : 1b01:fa06:0842:1510:fc:a41b:189a:31af

Règles d'écriture :

- on peut omettre les 0 de gauche de chacun des 8 entiers

1b01:fa06:0842:0:0:0:189a:31af → 1b01:fa06:842:0:0:0:189a:31af

- on peut remplacer *une seule fois* une suite de 0 consécutifs séparés par « : » par « :: »

1b01:fa06:842:0:0:0:189a:31af → 1b01:fa06:842::189a:31af

Adresse IPv6

L'adresse est codée sur 128 bits.

Représentation : 8 entiers hexadécimaux codés chacun sur 16 bits. Les entiers sont séparés par « : ».

Exemple : 1b01:fa06:0842:1510:fc:a41b:189a:31af

Règles d'écriture :

- on peut omettre les 0 de gauche de chacun des 8 entiers

1b01:fa06:0842:0:0:0:189a:31af → 1b01:fa06:842:0:0:0:189a:31af

- on peut remplacer *une seule fois* une suite de 0 consécutifs séparés par « : » par « :: »

1b01:fa06:842:0:0:0:189a:31af → 1b01:fa06:842::189a:31af

L'adresse **loopback** : ::1

Certaines adresses sont réservées à un usage spécifique :
adresses de préfixe :

- 127.0.0 : locale à l'hôte
- 10.0.0 ou 192.168 ou 172.16 à 172.31 : privée
- 224 à 239 : multidiffusion
- 255.255.255.255 : diffusion

Certaines adresses sont réservées à un usage spécifique :
adresses de préfixe :

- 127.0.0 : locale à l'hôte
- 10.0.0 ou 192.168 ou 172.16 à 172.31 : privée
- 224 à 239 : multidiffusion
- 255.255.255.255 : diffusion

notation CIDR (Classless Inter-Domain Routing) :

- permet de définir un ensemble d'adresses
- l'ensemble d'adresses 172.16 à 172.31 peut se noter 172.16/12. Cela signifie que les 12 premiers bits sont fixes et les 20 suivants libres. Il y a donc 2^{20} adresses possibles avec un préfixe de longueur 12 bits.

NAT (network address translation)

IPv4 réserve certaines adresses pour des usages privés :

- 10.0.0.0/8 : 10.0.0.0 \rightarrow 10.255.255.255
- 172.16.0.0/12 : 172.16.0.0 \rightarrow 172.31. 255. 255
- 192.168.0.0/16 : 192.168.0.0 \rightarrow 192.168. 255. 255

Pour communiquer avec l'extérieur, une entité avec une IP privée devra faire passer ses messages via un NAT.

adresse IPv6 locale au lien : adresses fe80 : :/10

Adressage IPv6

adresse IPv6 locale au lien : adresses fe80 : :/10

adresse IPv6 globale :

3 bits	45 bits	16 bits	64 bits
001	préfixe pour le routage global	sous-réseau	identification de l'hôte

adresses 2000 : :/3

Adressage IPv6

adresse IPv6 locale au lien : adresses fe80 : :/10

adresse IPv6 globale :

3 bits	45 bits	16 bits	64 bits
001	préfixe pour le routage global	sous-réseau	identification de l'hôte

adresses 2000 : :/3

- pas de correspondance automatique entre adresses IPv4 et IPv6
- les réseaux IPv4 et IPv6 cohabitent
- pour connaître la configuration réseau de votre machine :
commande `ip address show` ou `ifconfig` (obsolète)

Résolution de noms

Pour l'humain, difficile de retenir une adresse IPv4 ou IPv6.

→ utilisation de noms de domaine pour désigner une adresse IP

Exemples : **lucy** . **informatique.univ-paris-diderot.fr**
machine domaine

Commande pour connaître le nom et domaine de l'hôte : `hostname`

localhost désigne en général l'interface locale à l'hôte

serveur DNS (Domain Name System) : serveur permettant d'associer les adresses de noms avec leurs adresses IP.

- annuaire distribué (il y a donc plusieurs serveurs DNS)
- Le DNS contient aussi d'autres informations lié à un nom de domaine :
 - l'adresse IPV4 (A record)
 - l'adresse IPv6 (AAAA record)
 - les serveurs de courrier électronique pour le domaine (MX record)
 - les serveurs DNS du domaine (NS record)
- commandes : `host` et `dig`

protocole

Sur le réseau, les entités communiquent en suivant des règles rassemblées sous forme de **protocole**.

Il y a différentes sortes de protocoles. Des protocoles en charge :

- des formats des messages envoyés ou reçus par l'utilisateur
- des messages + adresses source et de destination
- des formats des données envoyées sur le réseau
- des données reçues et de leur cohérence

⇒ séparation en couches pour ces différents traitements.

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

6 Présentation

5 Session

4 Transport

3 Réseau

2 Lien

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

5 Session

4 Transport

3 Réseau

2 Lien

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

codage, cryptage des données (**data**)

5 Session

4 Transport

3 Réseau

2 Lien

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

codage, cryptage des données (**data**)

5 Session

connexion entre les deux hôtes (socket)

4 Transport

3 Réseau

2 Lien

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

codage, cryptage des données (**data**)

5 Session

connexion entre les deux hôtes (socket)

4 Transport

couper les données en **segments** de données,
les réassembler dans l'ordre (protocoles TCP,
UDP)

3 Réseau

2 Lien

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

codage, cryptage des données (**data**)

5 Session

connexion entre les deux hôtes (socket)

4 Transport

couper les données en **segments** de données,
les réassembler dans l'ordre (protocoles TCP,
UDP)

3 Réseau

transmettre les segments de données sous forme
de **paquets** - routage des paquets (protocole
IP)

2 Lien

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

codage, cryptage des données (**data**)

5 Session

connexion entre les deux hôtes (socket)

4 Transport

couper les données en **segments** de données,
les réassembler dans l'ordre (protocoles TCP,
UDP)

3 Réseau

transmettre les segments de données sous forme
de **paquets** - routage des paquets (protocole
IP)

2 Lien

transmettre physiquement les paquets sous
forme de **trames (frame)** au prochain noeud -
flux - détecter les erreurs de transmission

1 Physique

Les modèles en couches (layer)

Le modèle OSI (Open System Interconnection)

7 Application

interface utilisateur/applications réseaux
(navigateur, service mail, ...)

6 Présentation

codage, cryptage des données (**data**)

5 Session

connexion entre les deux hôtes (socket)

4 Transport

couper les données en **segments** de données,
les réassembler dans l'ordre (protocoles TCP,
UDP)

3 Réseau

transmettre les segments de données sous forme
de **paquets** - routage des paquets (protocole
IP)

2 Lien

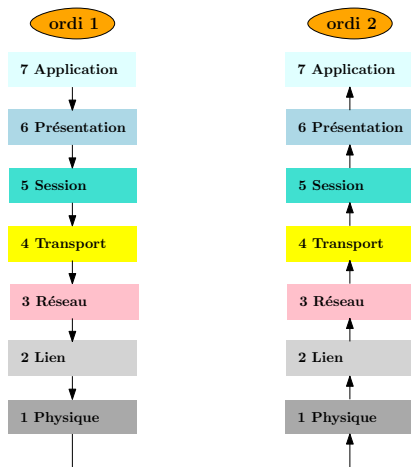
transmettre physiquement les paquets sous
forme de **trames (frame)** au prochain noeud -
flux - détecter les erreurs de transmission

1 Physique

paramètres des cable ethernet, Wifi, fibre
optique

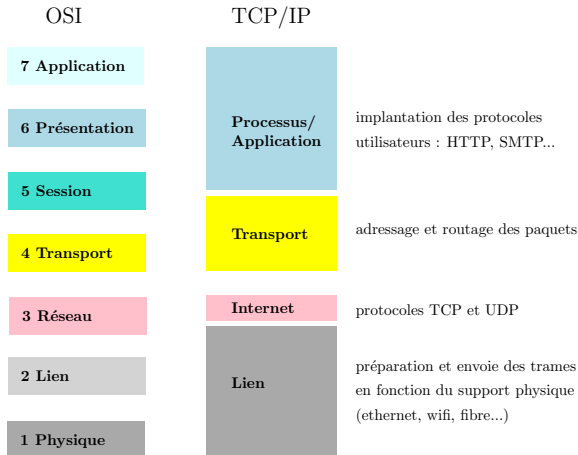
Les modèles en couches (layer)

Le modèle OSI



Les modèles en couches (layer)

Le modèle TCP/IP



Le modèle client/serveur

Des entités communiquent entre elles :

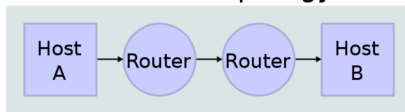
- une entité est le **serveur** qui répond aux requêtes des autres entités. Il fournit un **service**,
- chaque autre entité est un **client** qui entre en communication avec le serveur et l'interroge.

Le modèle pair à pair

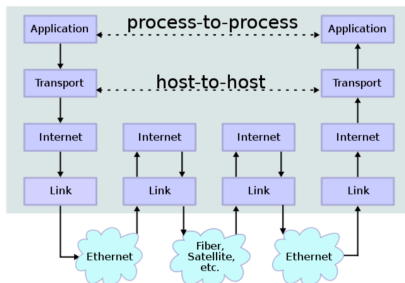
Toutes les entités ont le même rôle. Chaque entité est à la fois un client et un serveur.

Les modèles de communication

Network Topology



Data Flow



Source : Wikipedia

Requests For Comments (RFC) : documents officiels décrivant les standards, protocoles et technologies d'internet.

<https://www.ietf.org/standards/rfcs/>

exemples :

- RFC 2616 décrit le protocole HTTP/1.1
- RFC 2818 décrit le protocole HTTPS

Une machine peut avoir plusieurs applications réseaux qui tournent en parallèle.

De plus, ces applications peuvent utiliser la même adresse IP.

⇒ nécessité de distinguer les données entrantes.

⇒ **port** : entier non-signé de 16 bits qui correspond à une adresse locale à la machine.

adresse IP ↔ adresse d'un immeuble

port ↔ numéro d'appartement

Le numéro de port permet la différenciation des services lors de la réception.

Un couple (adresse, port) correspond à un **point de communication**.

- Les ports 0 à 1023 sont reconnus → correspondent à des services internet identifiés.
- Les ports 1024 à 49151 sont réservés → ceux que vous utiliserez. Également services d'usage moins général.
- Les ports 49152 à 65535 sont libres → durée limitée.

Correspondance entre le numéro de port et le service associé :

- sur le site de l'IANA (Internet Assigned Number Authority) :

[https://www.iana.org/assignments/service-names-port-numbers/
service-names-port-numbers.xhtml](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml)

- dans le fichier `/etc/services`

Simuler un serveur ou client simple

Les commandes `telnet` et `netcat` (ou `nc`) permettent entre autre de simuler un client simple.

```
telnet <adresse> <port>
```

```
netcat <adresse> <port>
```

Avec l'option `-l`, la commande `netcat` permet de simuler un serveur simple. Dans ce cas, on ne spécifie pas l'adresse.

Ordre des octets sur le réseau

Network Byte Order (ordre NBO)

Encodage des entiers sur une machine

- grand-boutiste ou big-endian (BE) : l'octet de poids fort sur la plus petite adresse mémoire

exemple : $2099202 = (00000000\ 00100000\ 00001000\ 00000010)_2$

entier sur 4 octets

→

0 ... 7	8 ... 15	16 ... 23	24 ... 31
00000000	00100000	00001000	00000010

Ordre des octets sur le réseau

Network Byte Order (ordre NBO)

Encodage des entiers sur une machine

- grand-boutiste ou big-endian (BE) : l'octet de poids fort sur la plus petite adresse mémoire

exemple : $2099202 = (00000000\ 00100000\ 00001000\ 00000010)_2$
entier sur 4 octets

→

0 ... 7	8 ... 15	16 ... 23	24 ... 31
00000000	00100000	00001000	00000010

- petit-boutiste ou little-endian (LE) : l'octet de poids faible sur la plus petite adresse mémoire

exemple : $2099202 = (00000000\ 00100000\ 00001000\ 00000010)_2$

→

0 ... 7	8 ... 15	16 ... 23	24 ... 31
00000010	00001000	00100000	00000000

Ordre des octets sur le réseau

Convention : les adresses IP et port sur le réseau sont encodés en big endian.

Sur votre machine, l'encodage des entiers peut être BE ou LE...

⇒ il faut penser à convertir adresse et port en big-endian avant envoi et se préoccuper de la conversion inverse à la réception.

Conversion big endian/codage hôte

En C, on peut faire appel aux fonctions suivantes de la bibliothèque `arpa/inet.h` :

<code>uint32_t htonl(uint32_t hostlong);</code>	host to network long
<code>uint16_t htons(uint16_t hostshort);</code>	host to network short
<code>uint32_t ntohl(uint32_t netlong);</code>	network to host long
<code>uint16_t ntohs(uint16_t netshort);</code>	network to host short