



Université  
De Boumerdes



Université  
De Limoges

**Département de physique/Infotronique  
IT/S5**

# **Couche Transport**

*Réalisé par* : Dr RIAHLA

Docteur de l'université de Limoges (France)

Maitre de conférences à l'université de Boumerdes



Université  
De Boumerdes



Université  
De Limoges

**Département de physique/Infotronique  
IT/S5**

# Introduction

# Introduction

## Rappel sur le concept des protocoles TCP-UDP

---

La couche transport contient deux protocoles permettant à deux applications d'échanger des données indépendamment du type de réseau emprunté

**Protocole TCP:** Fiable et orienté connexion, qui assure le contrôle et l'acheminement sans erreurs des paquets envoyés.

**Protocole UDP:** non fiable et sans connexion. Son utilisation présuppose que l'on n'a pas besoin ni du contrôle de flux, ni de la conservation de l'ordre de remise des paquets

# Introduction

## TCP vs UDP

TCP orienté connexion	UDP orienté non connexion
<ul style="list-style-type: none"><li>❖ Etablissement et maintien d'une connexion entre deux machines</li></ul> <p>Hé, tu m'écoutes ? Oui, je t'écoute J'ai le cours réseau OK Je vais te l'envoyer OK Tu l'as reçu ? Oui, c'est bon ! Salut Salut</p>	<ul style="list-style-type: none"><li>❖ Expédition des données</li></ul> <p>Je t'envoie le cours réseau</p> <ul style="list-style-type: none"><li>➤ Moins fiable que TCP</li><li>Contrôles d'erreurs, ordonnancement</li><li>➤ plus rapide</li></ul>



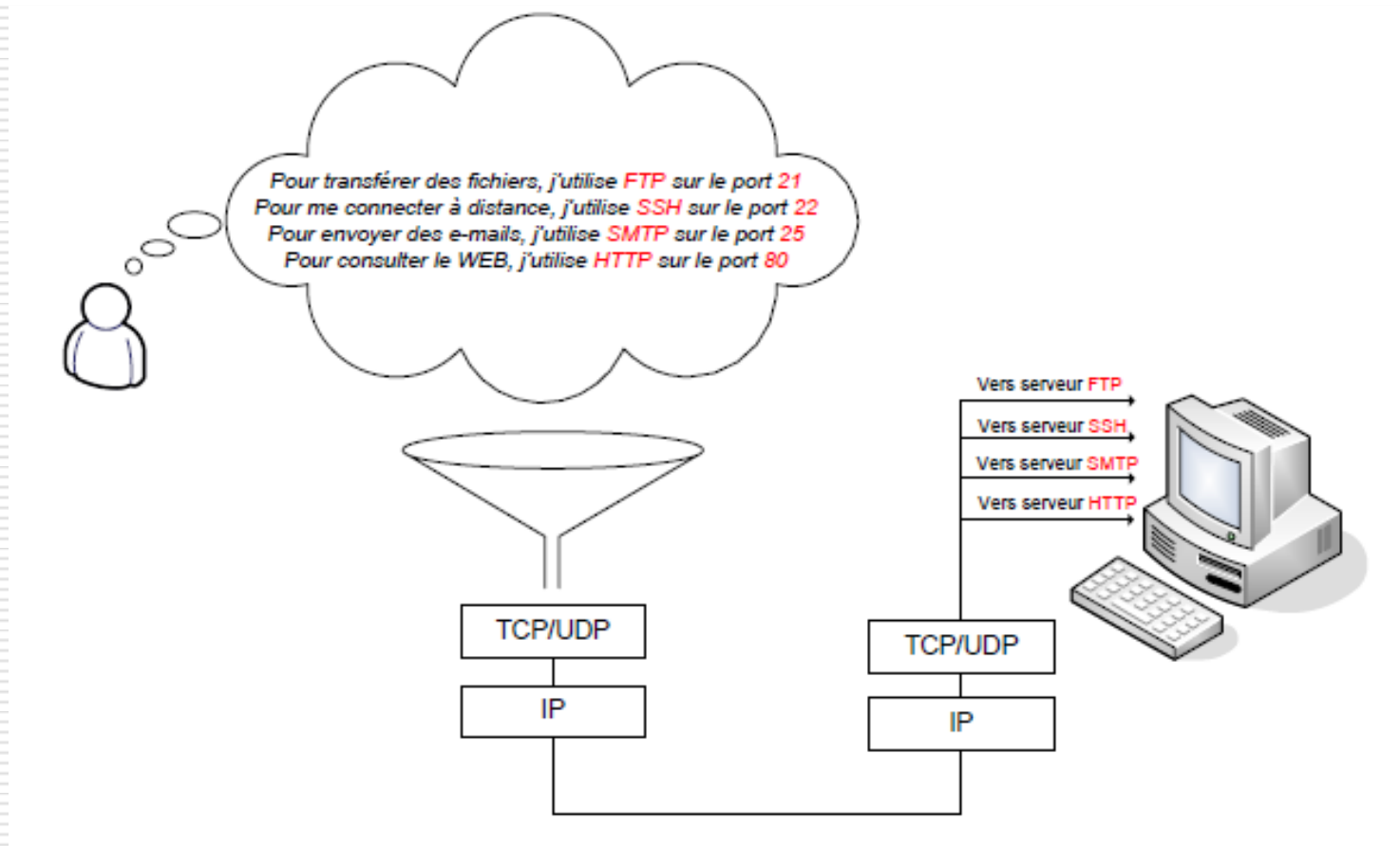
Université  
De Boumerdes



Université  
De Limoges

# Notion de ports

# Ports





Université  
De Boumerdes



Université  
De Limoges

# Message UDP

# Message UDP

## Le transfert UDP

---

- Un transfert UDP est caractérisé par :
  - IP source
  - IP destination
  - Port source
  - Port destination
- UDP multiplexe et démultiplexe les datagrammes en sélectionnant les numéros de ports
- Ne garantit la transmission des données
- Ne garantit l'ordre de transmission

<b>Port UDP source</b>	<b>Port UDP destination</b>
<b>Longueur message UDP</b>	<b>Somme de contrôle</b>
<b>Données ...</b>	





Université  
De Boumerdes



Université  
De Limoges

# Le segment TCP

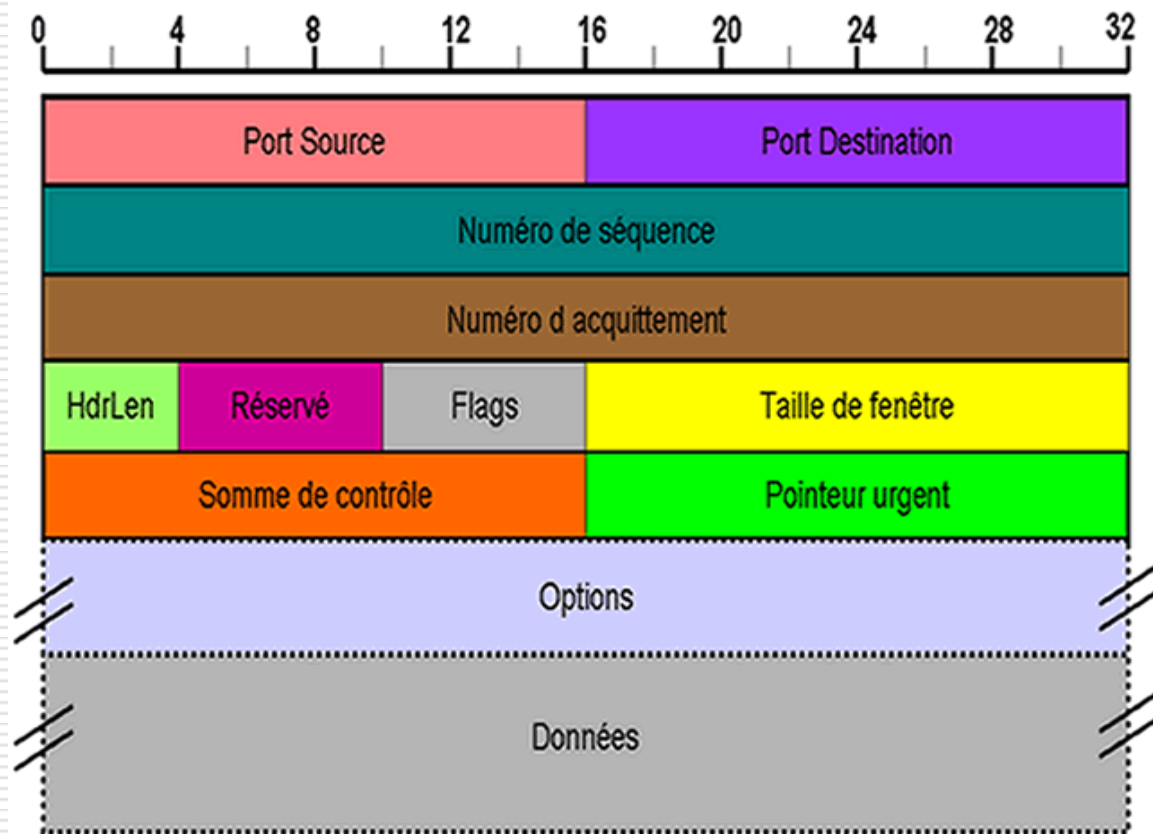
# Segment TCP

---

Le segment est l'unité de transfert du protocole TCP, il est utilisé pour les connexions en trois étapes:

- Établir les connexions.
- Transférer les données et émettre des acquittements.
- Fermer les connexions.

# Segment TCP



# Segment TCP

## Les champs du segment

---

- ❖ **Port Source** (16 bits): Port de l'application en cours sur la machine source
- ❖ **Port Destination** (16 bits): Port relatif à l'application en cours sur la machine de destination
- ❖ **Numéro de séquence (d'ordre) ISN**(32 bits): Numéro de séquence du premier octet de ce segment
- ❖ **Numéro d'acquittement** (32 bits): numéro d'acquittement est également appelé Le numéro d'accusé de réception correspond au numéro (d'ordre) du prochain segment attendu.

# Segment TCP

## Les champs du segment

---

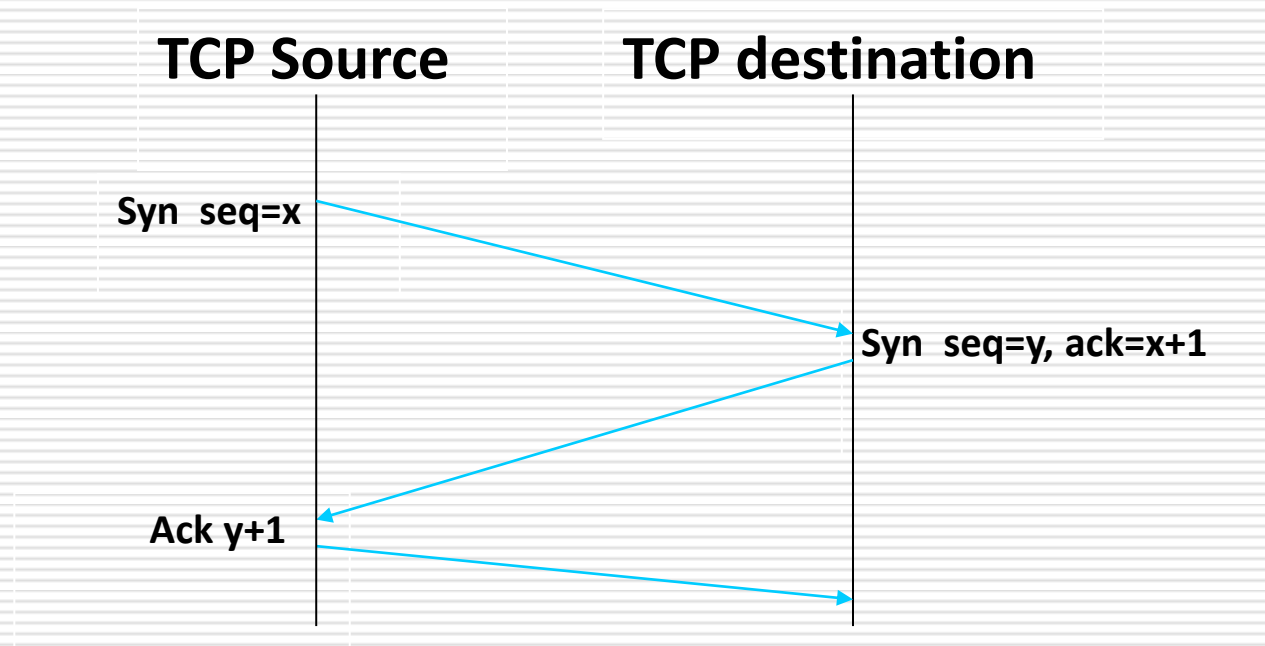
### ❖ Les flags

- **ACK:** Le paquet est un accusé de réception
- **FIN :** L'émetteur a atteint la fin de son flot de données.
- **RST:** Réinitialiser la connexion.
- **SYN:** Synchroniser les numéros de séquence pour initialiser une connexion.
- **PSH:** Fonction push.
- **URG :** Signale la présence de données URGentes.

# Segment TCP

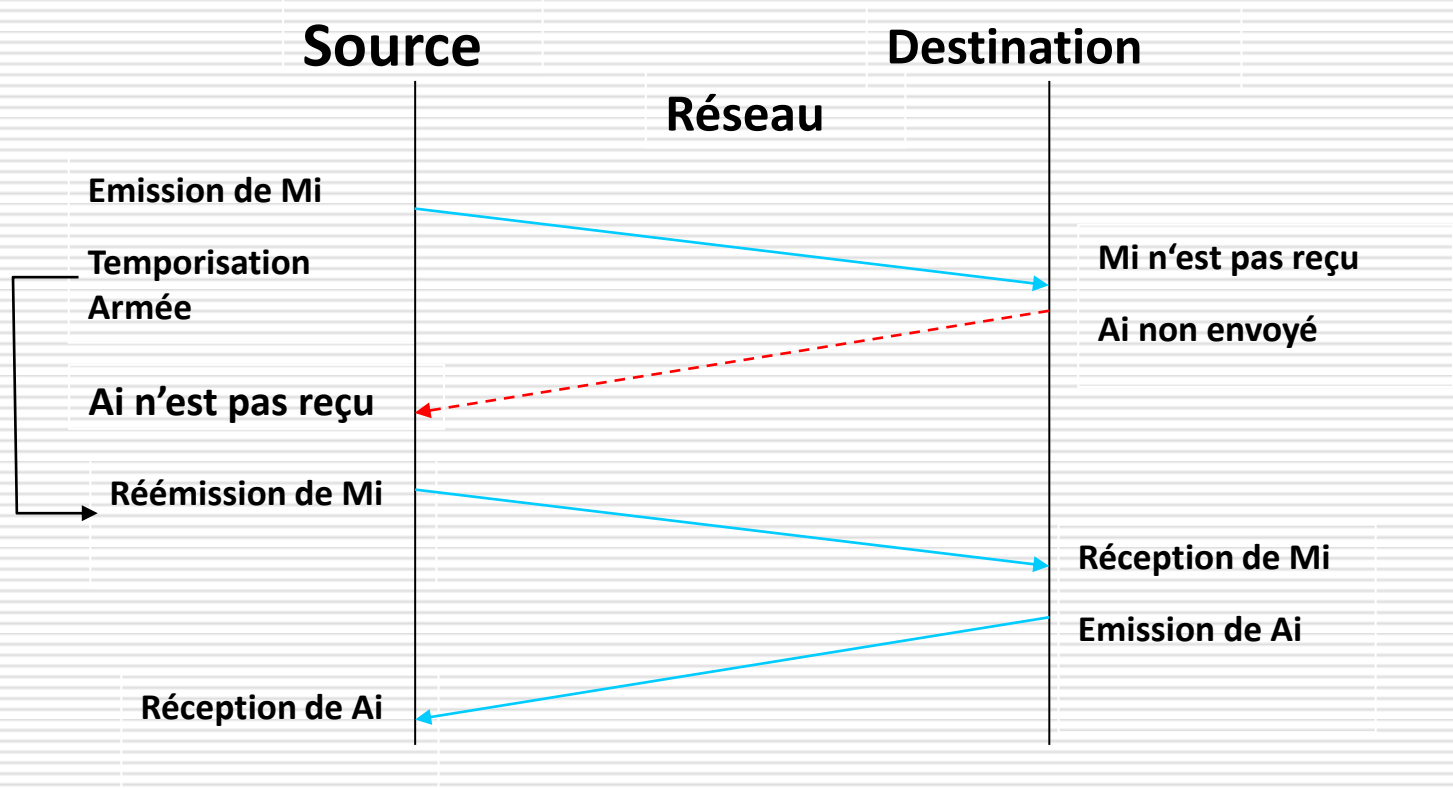
## Ouverture de connexion

---



# Segment TCP Acquittements

- Comment savoir si un paquet arrive ?
  - Accusé de réception



# Segment TCP

## Le fenêtrage

---

- Utilisation du réseau et gaspillage de bande passante

Envoi de données

Attente

Envoi d'acknowledge

Attente

...

- On peut faire mieux !

➤ fenêtrage



- Fenêtres TCP

Idée : prendre de l'avance sur les réponses c.à.d. fenêtre glissante

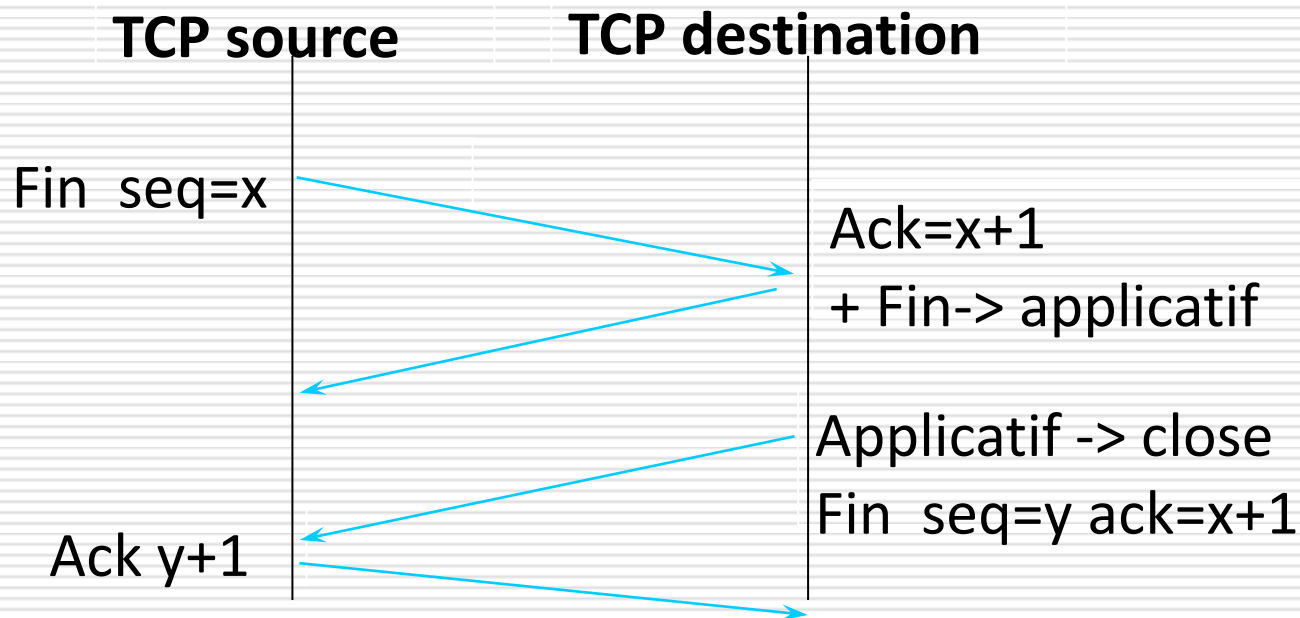




# Segment TCP

## Fermeture de connexion

La clôture de la connexion se passe de la même façon que l'ouverture, on utilise juste le flag FIN.



# Segment TCP

## Autres types de connexions

---

### **Transfert de données**

==> ACK=1 - SeqNum=101 - AckNum=301 - Data=30 octets

<== ACK=1 - SeqNum=301 - AckNum=131 - Data=10 octets

==> ACK=1 - SeqNum=131 - AckNum=311 - Data=5 octets

<== ACK=1 - SeqNum=311 - AckNum=136 - Data=10 octets

### **Fermeture brutale de connexion**

==> ACK=1 - RST=0 - SeqNum=200 - AckNum=400

<== ACK=0 - RST=1 - SeqNum=400 - ACKNum=xxx



Université  
De Boumerdes



Université  
De Limoges

# La connexion client serveur en deux modes

# Les sockets

---

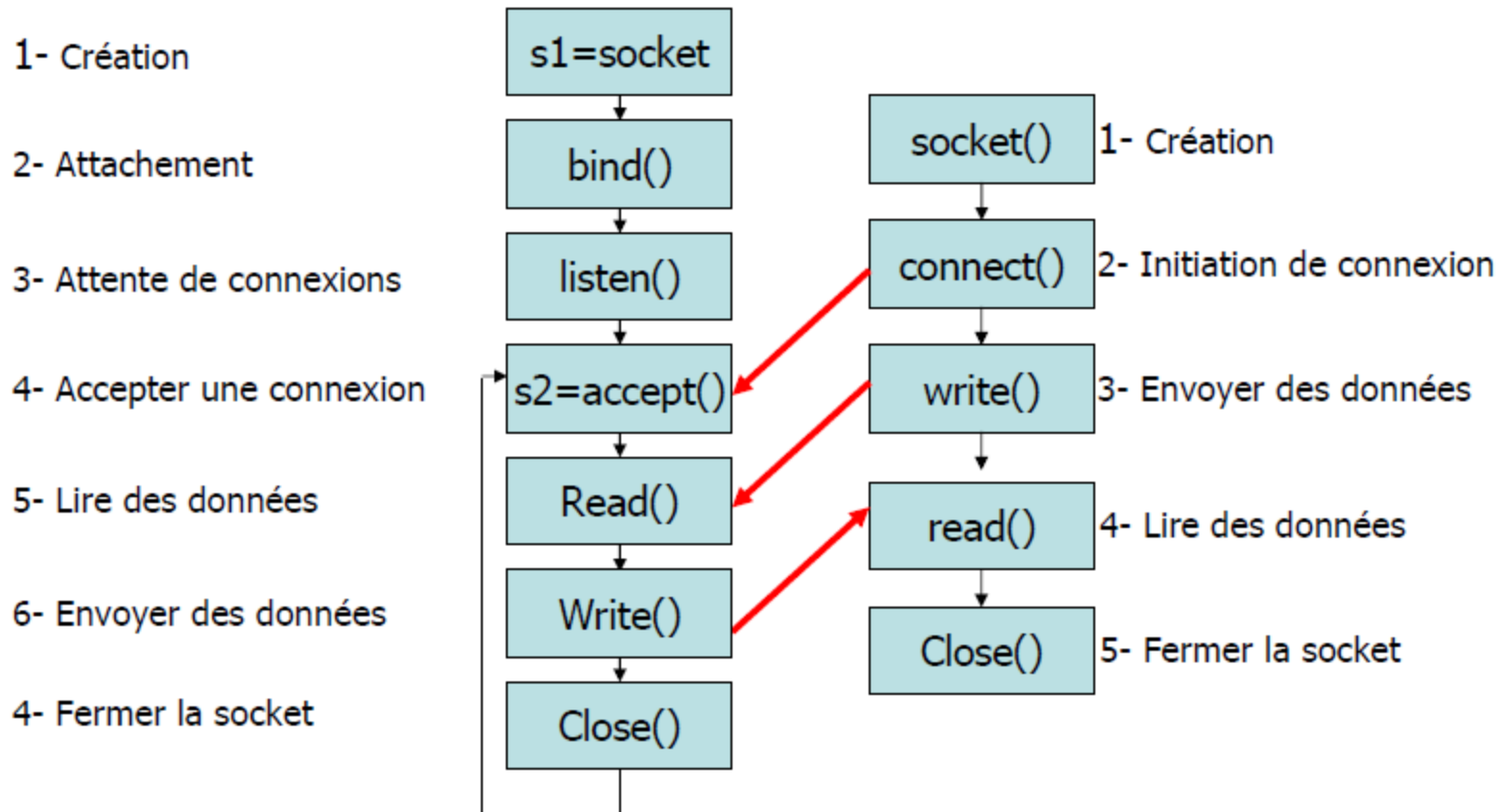
- C'est une interface réseau programmable avec un langage de programmation qui permet à divers processus de communiquer à travers un réseau TCP/IP.
- Dans Unix/Linux, un socket est comme un descripteur fichier sur lequel on effectue des opérations de Lecture/Ecriture selon les besoins.
- Les socket utilisent les différentes familles de protocoles réseaux et les protocoles de transport TCP et UDP.

# Communication Client/serveur

---

- Chacun des clients et du serveur doivent créer un socket avec l'utilitaire des fonctions d'un langage donné (JAVA ou C).
- Le serveur initialise l'adresse IP et le numéro de port du socket sur lequel il va écouter les requêtes entrantes ainsi que le client initialise l'adresse IP et le numéro de port du socket du serveur qu'il va contacter.
- Il existe deux modes de communication des sockets client/serveur. Le mode connecté (TCP) et le mode non connecté (UDP)

# Communication Client/serveur en mode connecté TCP



# Communication Client/serveur en mode non-connecté UDP

1- Création

s1=socket

2- Attachement

bind()

3- Attente de messages

recvfrom()

3- Envoie de messages

sendto()

4- Fermeture de la socket

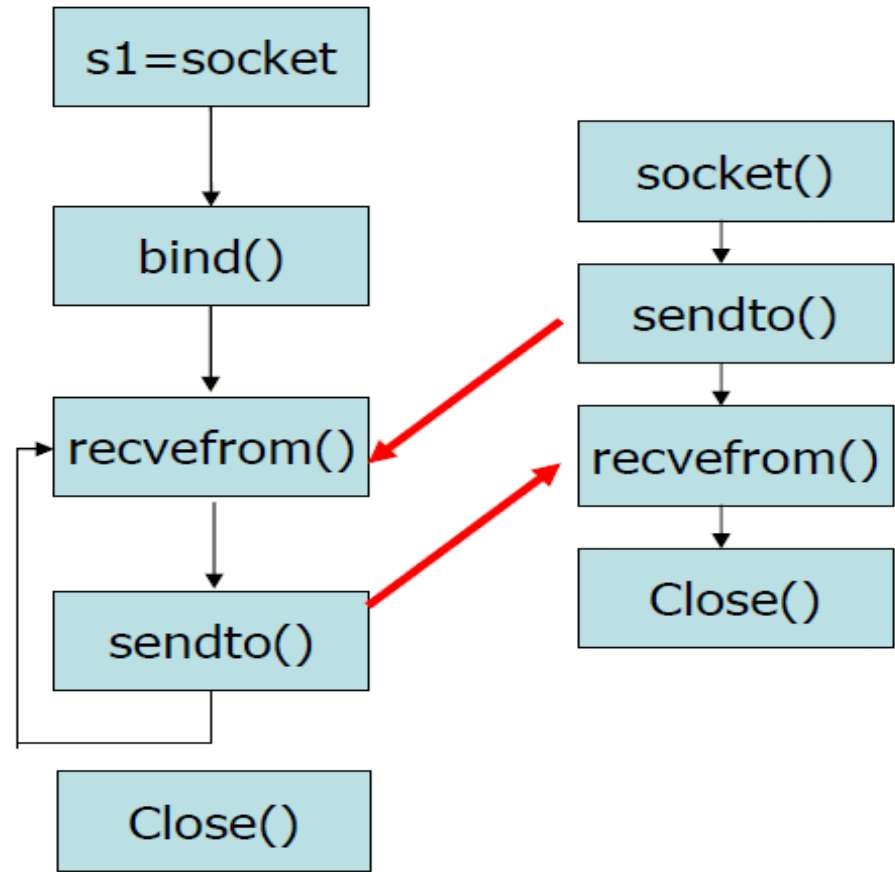
Close()

socket()

sendto()

recvfrom()

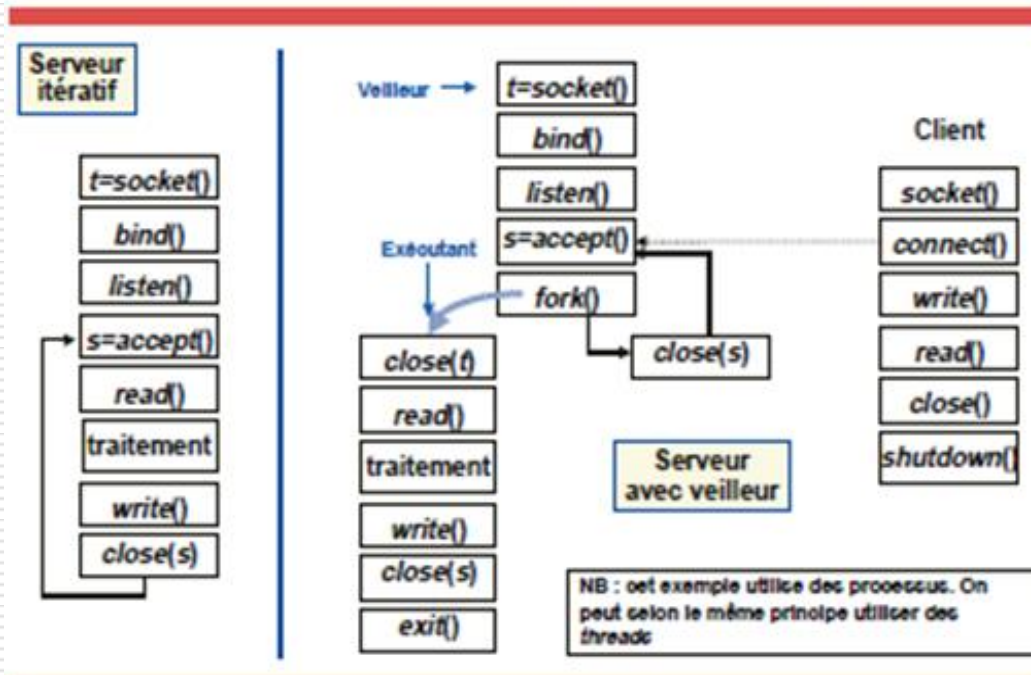
Close()



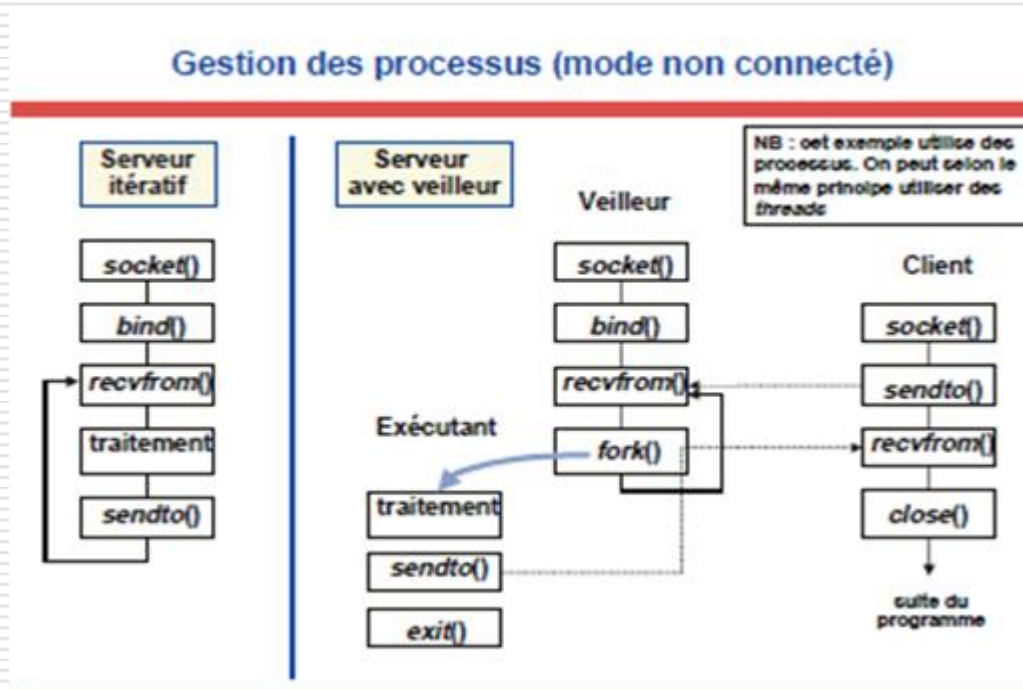


# Processus mode connecté

## Gestion des processus (mode connecté)



# Processus mode non connecté



# Buts

---

Différencier entre les modes de transport orientés connexion et non-orientés connexion

Comprendre l'utilité des numéros de ports utilisés dans TCP & UDP

Utiliser les sockets pour écrire des applications réseaux au dessus de la pile de protocoles fournie par le système d'exploitation

Comprendre la difficulté d'écrire des applications réseaux en utilisant les sockets

---



Université  
De Boumerdes



Université  
De Limoges

# **Vous voulez programmer RESEAU!!!**





Université  
De Boumerdes



Université  
De Limoges



C'est Très simple

# Ecrire des applications réseaux en utilisant les sockets

---

