CHAPITRE 6

La couche transport

Les protocoles au niveau de la couche transport

•La couche transport définit deux protocoles de transport (TCP et UDP) qui s'exécutent au-dessus de IP (la couche réseau).

•La couche transport utilisent les services de IP pour transporter les données .

•Le Protocole TCP (Transmission Control Protocol) : orienté connexion , fiable , renvoie toute donnée non reçue → mais il est lent.

•Protocole UDP (User Datagram Protocol) : orienté non connexion, peu fiable , n'utilise pas d'accusés de réception et n'assure aucun contrôle de flux → mais il est rapide.

Notion de port

- Un port est un concept **abstrait** (pas physique) , c'est un numéro qui permet **d'identifier une application sur une** machine.

- On dit qu'une application est **attaché à un port**.

- Chaque machine attache des numéros de port à ses applications **indépendamment des autres machines**.

- Deux applications sur la même machine **ne peuvent pas avoir le même port** .

Numéro de port

- Le numéro de port est sur **16 bits**.
- 65 535 valeurs différentes → insuffisant pour identifier toutes les applications de tous les machines sur un réseau.
- les ports n'ont pas une **signification globale** → une signification restreinte à une machine.
- Le triplet (**protocole , numéro de port, adresse IP hôte**) permet d'identifier d'une manière unique une application.

Modèle de communication client/serveur

- Le **modèle de communication** utilisé entre les applications est le modèle **client/serveur**.

- Un serveur est **un programme (application)** qui est toujours en cours d'exécution , en attente des requêtes des clients.

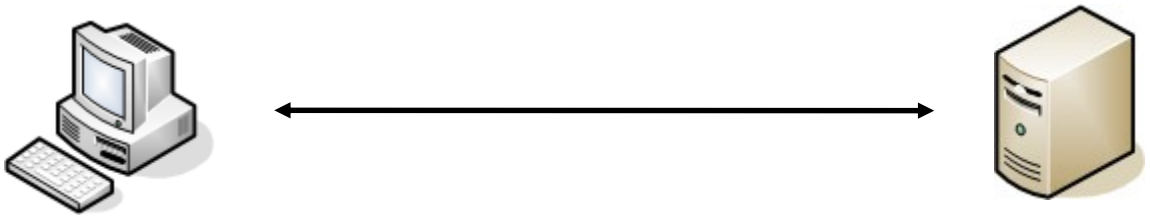
- On parle plus **de service** que **de serveur** pour ne pas le confondre avec le serveur au sens machine → **plusieurs services** peuvent fonctionner sur la même machine.

- Un client est un programme qui communique avec un serveur (il envoi des requêtes)

Exemple :

un client web (navigateur) communiquent avec un serveur web.

Modèle client /serveur
exemple 1 de communication



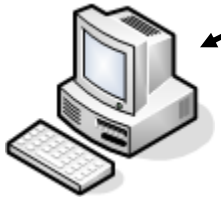
Client
192.168.1.2
Application sur le Port 1400

Le client doit se connecter au serveur
via l'adresse du serveur ainsi que le port
communiquer (192.168.1.20, port 80)

Serveur
192.168.1.20
Application sur le Port 80

Une fois la connexion est effectuée ,
le serveur connaît le port du client
Possibilité d'échangé des données

Modèle client /serveur
exemple 2 de communication



Client
192.168.1.2
Port 1400
communiquer (192.168.1.20, port 80)

Serveur
192.168.1.20
Application Port 80
Application port 21

Client
192.168.1.5
Port 1400

Communiquer (192.168.1.20, port 21)

Catégories de ports

- Il existe 3 catégories de ports :

- **ports connu (well-known) :** de 0 à 1023 ne peuvent être utilisés que par des processus système ou des programmes exécutés par des utilisateurs privilégiés.

- **ports registered :** de 1024 à 49 151 peuvent être utilisés par des processus utilisateur ordinaires ou des programmes exécutés par des utilisateurs ordinaires

- **ports dynamic/private :** de 49 152 à 65 535 alloués dynamiquement aux applications clients

Quelques Ports UDP connus

Numéro	Nom	Description
53	DNS	Domain name server Résolution de Nom
67	BOOTPS	Boot protocol server
68	BOOTPC	Boot protocol client
69	TFTP	Trivial file transfert protocol
123	NTP	Network time protocol
161	SNMP	Simple network management protocol
520	RIP	Utilisé par le protocole RIP pour l'échange des tables de routage

Quelques Ports TCP

Numéro	Nom	Description
21	FTP	File Transfer Protocol pour l'échange de fichiers
22	SSH	Secure Shell Pour le travail à distance sécurisé
23	TELNET	telnet Pour le travail à distance
25	SMTP	Simple mail transfer pour l'envoi d'un courrier électronique
37	TIME	time
53	DNS	Domain name server
80	HTTP	Hyper-Text transefer ptotocol pour la consultation d'un serveur HTTP par le biais d'un Navigateur web
110	POP3	Post office protocol version 3 , pour la récupération de son courrier électronique

Quelques Ports TCP

Numéro	**Nom**	**Description**
500	IPSEC	500, port utilisé pour le canal d'échange de clés IPsec
1723	PPTP	, pour l'utilisation du protocole de VPN PPTP
3306	MySQL	*Le SGBD MySQL .*
1433	SQLServer	Le SGBD SQL server
5432	*postgresql*	*Le SGBD postgresql*

UDP : Caractéristiques

- Mécanisme d'identification des processus d'applications avec un numéro de port.
- N'est pas orienté connexion → pour envoyer des données , on doit pas ouvrir une connexion.
- Ne séquence pas les données.
- Plus rapide, plus simple, plus efficace → mais moins fiable (risque de perte).

UDP : Fonctionnement

- Les données UDP plus l'en-tête sont encapsulés dans un seul datagramme IP.
- Les couches UDP et IP sont très liées car UDP a besoin de connaître les adresses IP source et destination.
- UDP va construire le datagramme et le passer à IP .
- IP va l'envoyer sans , pratiquement , aucun contrôle.

UDP : Format de l'entête

Nombre de bits				
16	16	16	16	?
Port source	Port destination	Longueur	Total de contrôle	Données

- Port source = numéro de port du processus émetteur.
- Port destination = numéro de port du processus de destination.
- Longueur = longueur du paquet UDP en octets. (En-tête + données)
- Total de contrôle = permet de vérifier l'intégrité des données. Si ce champ est à 0, c'est que l'on utilise pas de vérification d'intégrité.
- Taille de l'entête = 8 octets

TCP : caractéristiques

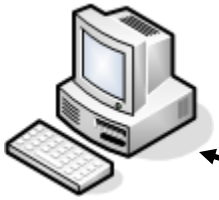
- Mécanisme d'identification des processus d'applications → en utilisant le principe des ports .
- Orienté connexion → avant d'envoyer les données on doit ouvrir une connexion entre l'émetteur et le récepteur.
- Garantie de remise de messages dans l'ordre de l'émission → Utilise des numéros de séquences pour ordonner les segments de données.
- Utilise les acquittements pour acquitter les segments.
- Fiable mais lent par rapport à UDP.

Identification d'une connexion

- Avant de pouvoir envoyer des données, on doit d'abord établir une connexion entre l'émetteur et le récepteur (les extrémités de la connexion).
- Une extrémité est définie par une adresse IP et un numéro de port.
- Une connexion est définie par le protocole et les deux extrémités.

  (protocole de transport, @IP1, Num port1 , @IP2, Num port2)
- Le protocole de transport n'apparaît qu'une fois car c'est le même aux deux extrémités.

Exemple pour identifier une connexion



Client
192.168.1.4
Port 1400

Serveur
192.168.1.1
Port 21

Deux extrémité : (192.168.1.4 , 1400) et (192.168.1.1, 21)

Deux extrémité : (192.168.1.5 , 1500) et (192.168.1.1, 21)

Client
192.168.1.5
Port 1500

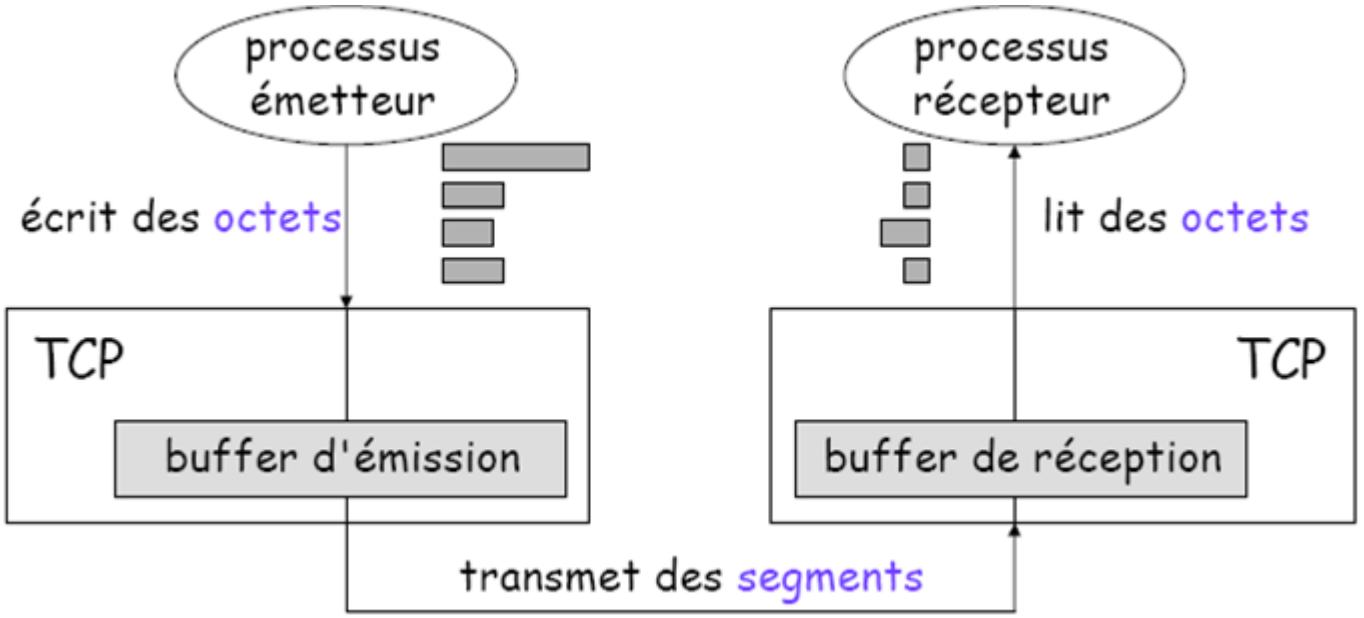
Une connexion par : protocole, extrémité 1, extrémité 2

(tcp, 192.168.1.4, 1400, 192.168.1.1, 21)

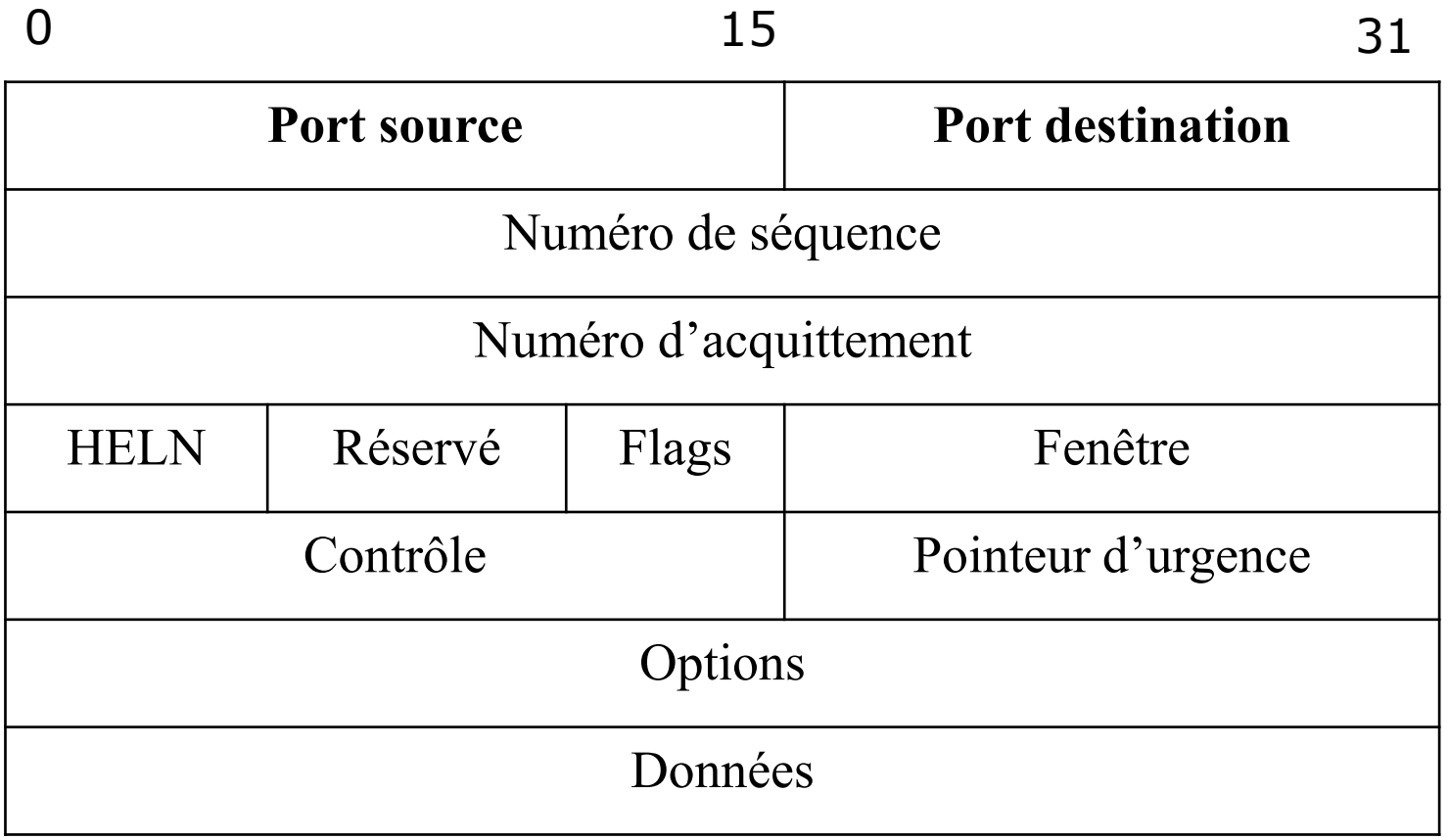
(tcp, 192.168.1.5 , 1500, 192.168.1.1, 21)

Fonctionnement de TCP

- Les applications (processus) envoient leur données au protocole TCP sans se préoccuper de la taille.
- Les données reçues sont gérées comme un flux d'octets (stream)
  → TCP , ne s'intéresse pas a leur signification .



Format de l'entete TCP



- **Port source et destination (sur 16 bits chacun):** identifier les extrémités de le connexion TCP.
- **N° de séquence (sur 32 bits):** numéro du premier octet des données émis.
- **N° d'accusé réception (numéro d'acquittement sur 32 bits)** : numéro du dernier octet reçu + 1, donc le numéro du prochain octet à recevoir.
- **HLEN (4 bits)** : contient le nombre de mots de 32 bits dans l'en-tête.
- **Réservé (6 bits)** : contient uniquement des 0.
- **Fenêtre** : le récepteur fixe la taille de la prochaine "fenêtre" pour l'émetteur. C'est le nombre d'octets qu'il est en mesure d'accepter à partir du numéro d'acquittement.
- **Checksum** : total de contrôle.
- **Pointeur d'urgence** : si le drapeau URG vaut 1, ce pointeur indique le numéro de séquence du dernier octet urgent.
- **Options** : permet de préciser certaines options TCP (ex: la taille des segments).
- **Données :** données de la couche supérieure (couche application).

Les falgs (indicateurs)

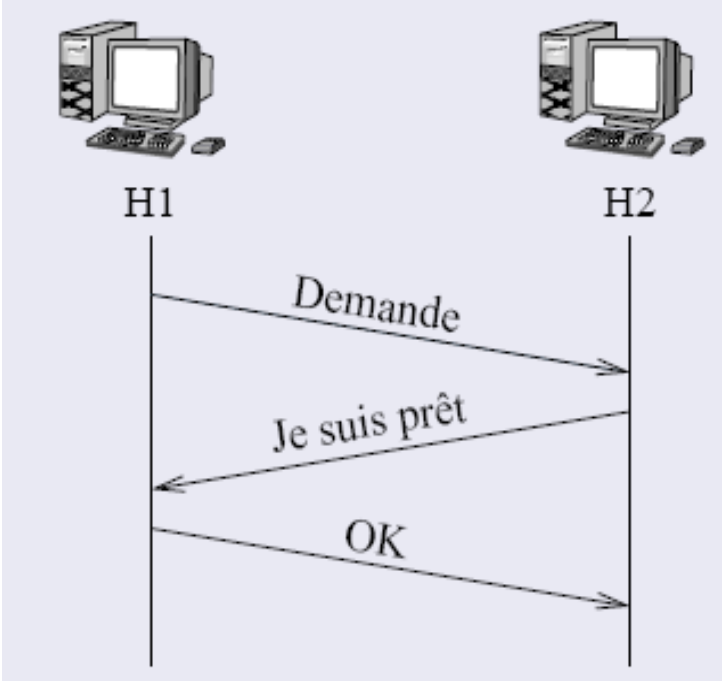
Les flags (indicateurs)					
URG	ACK	PSH	RST	SYN	FIN

- Il existe 6 indicateurs ,qui sont utilisés pour l'ouverture et la fermeture de la connexion.
- **Flag ACK** : vaut 1 si le champ "Numéro d'acquittement" est valide.
- **Flag SYN** : vaut 1 lors d'une ouverture de connexion .
- **Flag FIN** : vaut 1 lors d'une fermeture de connexion.
- **Flag RST** : vaut 1 pour une réinitialisation da la connexion à cause d'erreur irrécupérable. Ce n'est pas la procédure normale pour arrêter une connexion TCP.
- **Flag PSH** : vaut 1 pour indiquer une remise immédiate des données au processus de la couche supérieure.
- **Flag URG** : vaut 1 pour indiquer l'envoi des données urgents.

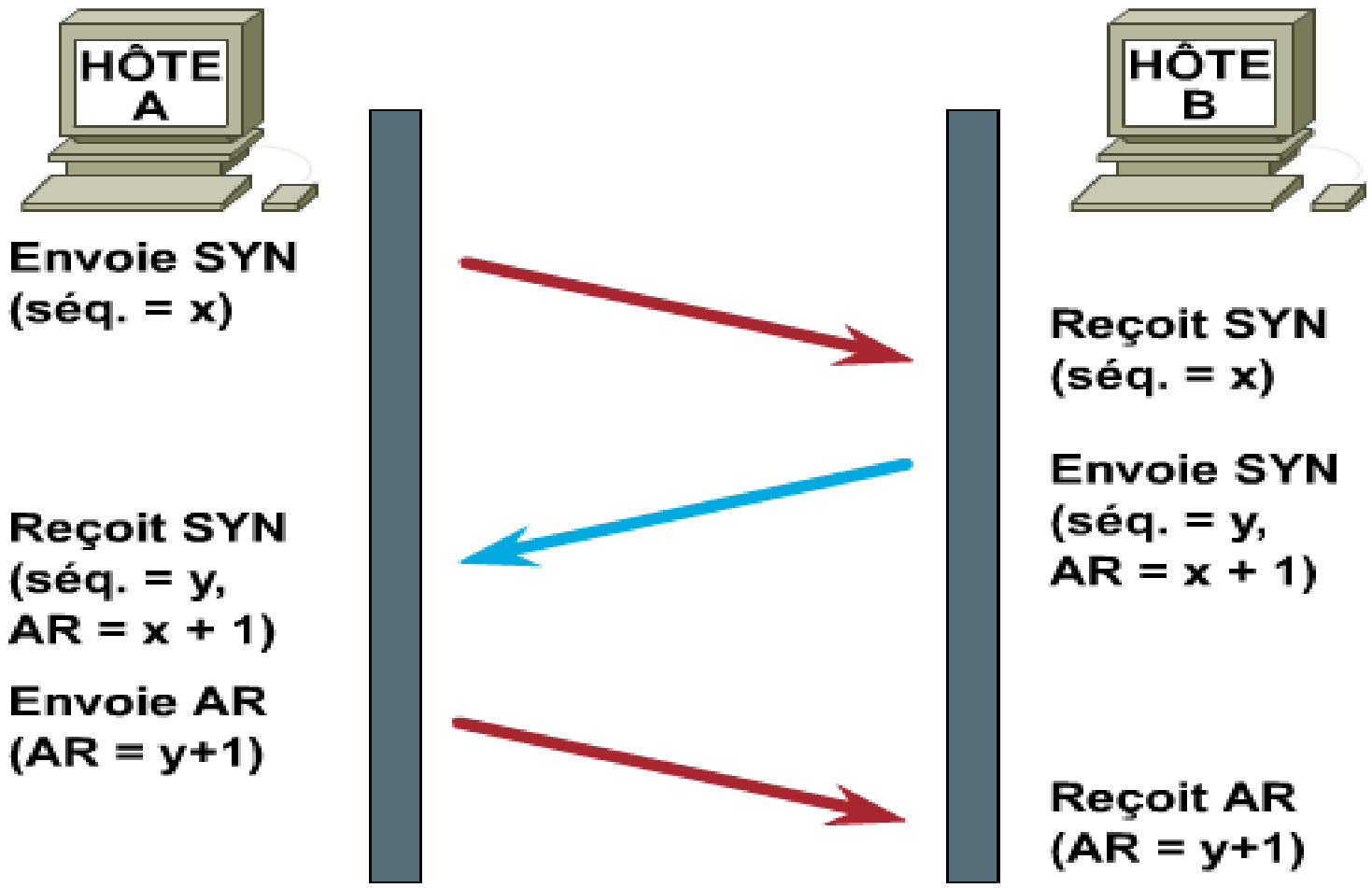
Ouverture d'une connexion

- La connexion se fait en 3 étapes :

- Emetteur : demande de connexion
- Récepteur : confirmation
- Emetteur : Confirmation générale



TCP : Ouverture d'une connexion



Les accusés de réception

- TCP garantit l'arrivée des messages, c'est-à-dire qu'en cas de perte, les deux extrémités sont prévenues.
- Ce concept repose sur les techniques d'acquittement de message : lorsqu'une source S émet un message Mi vers une destination D, S attend un acquittement Ai de D avant d'émettre le message suivant Mi+1.
- Si l'acquittement Ai ne parvient pas à S, S considère au bout d'un certain temps (time out) que le message est perdu et réémet Mi.
- C'est possible de faire des acquittement groupé → acquitter plusieurs segment en utilisant un seul acquittement
- Un numéro de séquence est attribué à chaque octet transmis et requiert un acquittement positif (ACK).
- Lors de la réception, les numéros de séquence servent à ordonner les segments et à éliminer les doublons.