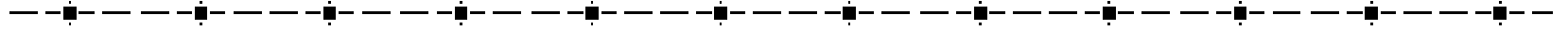




La couche Transport et les applications

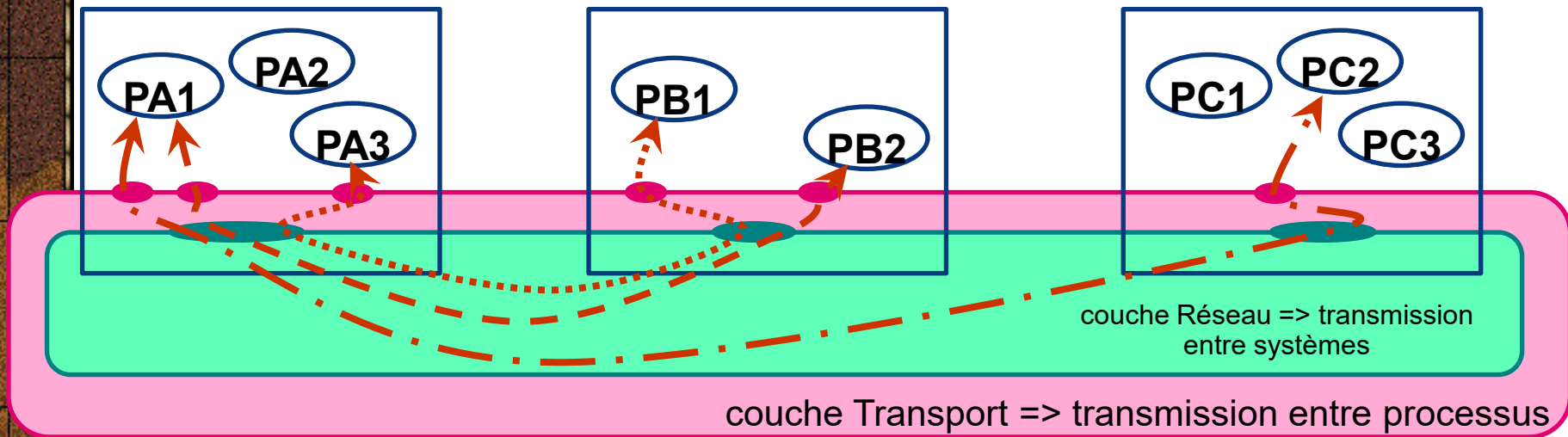
Protocole TCP / UDP
Quelques applications



La couche transport

Couche 4 – Généralité (1)

Le rôle de la couche Transport est d'acheminer des octets entre **des processus** s'exécutant dans des systèmes appartenant à la même couche Réseau, sans qu'ils aient à se préoccuper des problèmes de médium, de routage, etc...



Couche Transport : communication de bout en bout

- Couche basse : couche 1 à 4
- Couche haute : couche 4 à 7

Couche 4 – Généralité (2)

□ Service fourni

- ◆ transmission de données entre processus
- ◆ données normales, données express
- ◆ Transparence (vis-à-vis de la couche supérieure)
- ◆ mode connecté, mode non connecté
- ◆ Encapsulation des données de la couche supérieure

□ Les couches Transport les plus utilisées

- ◆ ISO
 - 1 Service en mode connecté, 5 classes de protocole
 - 1 Service en mode non connecté, 1 protocole
- ◆ technologie TCP/IP
 - 1 Service en mode connecté (**TCP**), 1 protocole
 - 1 Service en mode non connecté (**UDP**), 1 protocole

Classe protocole transport

	Transport ISO					
	mode connecté					mode non connecté
	0	1	2	3	4	
connexion	oui	oui	oui	oui	oui	non
transfert de données	oui	oui	oui	oui	oui	oui
correction des erreurs signalées	non	oui	non	oui	oui	non
contrôle de flux	non	non	oui	oui	oui	non
données express	non	non	oui	oui	oui	non
détection et correction des erreurs non signalées	non	non	non	non	oui	non
multiplexage	non	non	oui	oui	oui	non
éclatement sur plusieurs connexions Réseau	non	non	non	non	oui	non

technologie TCP/IP	
TCP	UDP
oui	non
oui	oui
oui	non
oui	non
oui	non
oui	non
non	non
non	non

Classes (de qualité) de Service Réseau

	A	B	C
détecte les erreurs	oui	oui	non
taux d'erreurs détectées	fort	fort	nul
tente de corriger les erreurs détectées	oui	non	non
taux d'erreurs corrigées	fort	faible	nul
signale les erreurs non corrigées	oui	oui	non

TCP / UDP

TCP Service en mode connecté

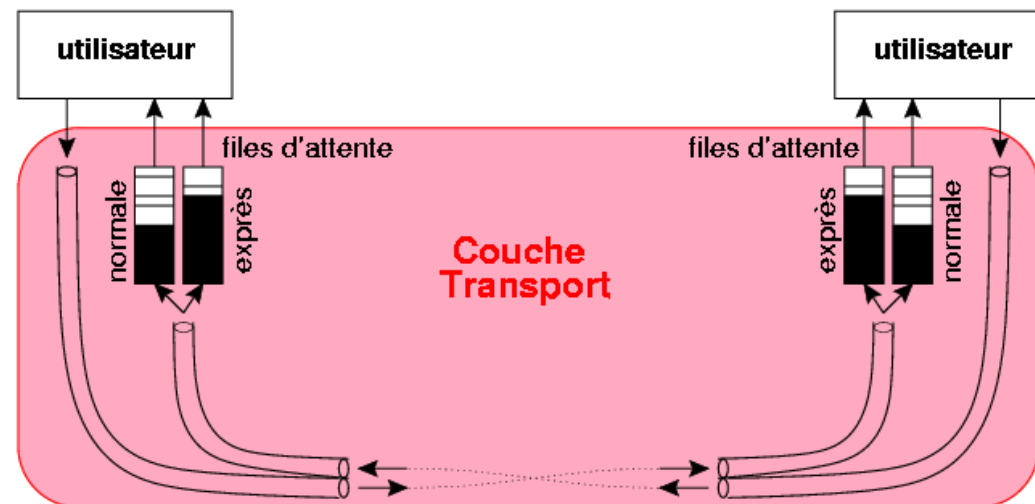
- jusqu'à 65534 points d'accès par système
- détection et tentative de correction des pertes et duplications
- signalement des erreurs
- livraison dans l'ordre d'émission
- contrôle de flux
- **Service fiable**

UDP Service en mode non connecté

- jusqu'à 65534 points d'accès par système
- un ou plusieurs destinataires
- **Service "non fiable"**

TCP

- ◆ mode connecté → canaux de transmission indépendants
- ◆ files d'attente des octets en attente de livraison
- ◆ terminaison de connexion = destruction des canaux
- ◆ **3 phases**
 - ◆ Établissement de la communication
 - ◆ **Triple poignée de main (Three way handshake)**
 - ◆ Transmission des données
 - ◆ Terminaison de la communication



Point d'accès aux services TCP/UDP

- Le concept de point d'accès au Service TCP (resp. UDP) est traduit par le concept **de socket** des systèmes d'exploitation
- **La socket** : lien entre l'applicatif et la couche transport
- *Une socket par communication.*
- création de *socket* selon les besoins des processus
 - par la fonction **socket ()**
 - une *socket* appartient à un processus
- attribution d'une adresse de point d'accès (**numéro de port**)
 - par la fonction **bind ()** (implicite ou explicite)

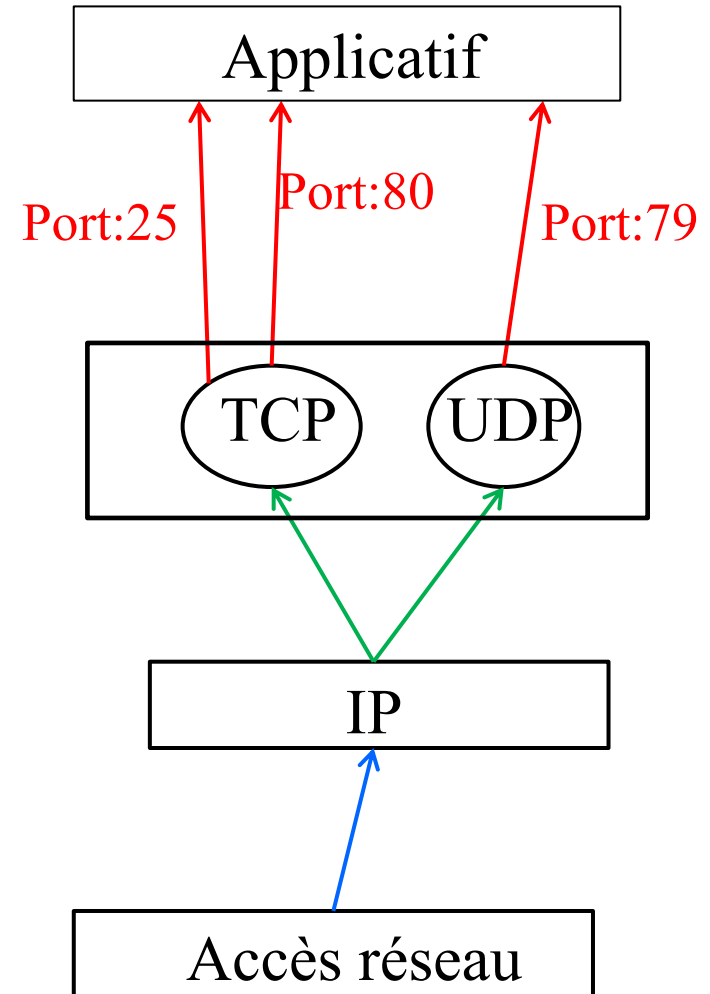
Port d'une connexion (1)

✦ Identifier un processus sur une machine

◆ 3 informations obligatoires :

- L'adresse IP
- Le protocole utilisé
- Le numéro de port
 - ◆ Privilégié
 - ◆ Bien connu
 - ◆ enregistré

Certaines informations sont implicites ou configurables pour certaines applications



Port d'une connexion (2)

✦ Port applicatif

- ◆ Unicité d'un port / applicatif
- ◆ **Privilégié (≤ 1024)**
 - 22 : ssh , 23 : telnet , 80 : http, 110 : POP3, 143: IMAP...
- ◆ Connu > 1024
 - 8080 : alternate http, 6000 : serveur X, 3724 : warcraft,..
- ◆ Dynamique ou privée ≥ 49152
- ◆ Plusieurs états d'un port en TCP:
 - Inexistant : aucune communication possible
 - **LISTEN** : en attente d'une connexion
 - **ESTABLISHED** : en communication
 - TIME_WAIT : fermeture de la connexion

Transmission de données

Les données sont transmises dans des PDU-TCP (resp.PDU-UDP), transportées par le Service IP qui livre ... peut-être, une seule fois

Les risques :

- ◆ perte de PDU-IP → perte de la PDU-TCP ou PDU-UDP encapsulée et des données qu'elle contient
- ◆ livraisons multiples de la même PDU (duplication), donc de données
- ◆ livraison d'une suite d'octets dans un ordre différent de celui de la soumission

UDP ne fait ni détection, ni correction

TCP tente de détecter tous les problèmes et de les corriger

Fiabilité de TCP (1)

➤ Numérotation lors de l'expédition

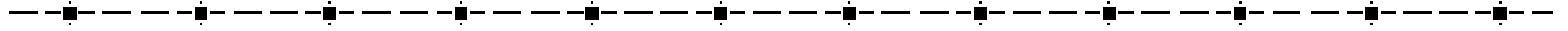
- ◆ les octets de l'utilisateur sont transmis par bloc dans des PDU-TCP
- ◆ l'entité-TCP calcule un numéro de séquence qui correspond "aux nombres d'octets envoyés" et le place dans la PDU envoyé, cela correspond au numéro **seq** : séquence
- ◆ **initialisation de seq** : à l'établissement de connexion
- ◆ Calcul lors de la phase d'initialisation d'un nombre correspondant "aux nombres d'octets reçus" (aussi envoyé dans la PDU), **c'est le numéro ack** : acquittement

➤ Vérification lors de la réception

- ◆ $n^\circ \text{ reçu} > n^\circ \text{ attendu}$ → une perte
- ◆ $n^\circ \text{ reçu} < n^\circ \text{ attendu}$ → une duplication
- ◆ $n^\circ \text{ reçu} = n^\circ \text{ attendu}$ → OK

➤ Correction

- ◆ *duplication* : l'entité-TCP ignore la PDU-TCP en doublon
- ◆ *perte* : retransmission de la PDU contenant le bloc
 - ◆ on renvoie la partie perdue



Applicatifs Généralités

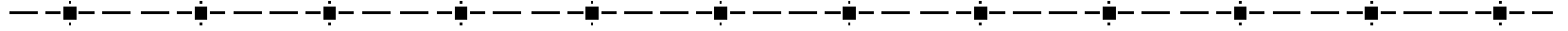
Généralité

✦ Notion client/serveur

- ◆ Serveur : en attente de connexion → un port ouvert
- ◆ Client : veut une information détenue par le serveur
 - *Connait le nom ou l'@IP du serveur*
 - Le numéro de port
 - La requête à effectuer

✦ Client

- ◆ processus créé lorsque le besoin apparaît
 - par une personne
 - au démarrage du système
- ◆ l'entité-client doit savoir à quelle entité-serveur elle doit/peut s'adresser
 - au lancement du processus : fichier de configuration, argument
 - par saisie pendant l'exécution



Quelques exemples d'Applicatifs

DHCP : définitions

DHCP : Dynamic Host Configuration Protocol -> RFC 2131

(successeur de BOOTP)

utilisation de la couche transport UDP : port 67 client -> serveur


port 68 serveur -> client

- Utilisation pour récupération automatique @IP, etc... sur un ordinateur

Serveur fournit (en général) :

- Adresse IP
- Masque de réseau
- Adresse de la passerelle
- Adresse du DNS, nom du domaine

Un serveur a un « pool » d'adresses qu'il peut distribuer

 utilisation d'un **bail** sur la durée de location d'une adresse

DHCP : fonctionnement

1. Emission d'un broadcast pour la demande d'une adresse IP
-> *DHCPDISCOVER*
2. Réponse en unicast d'un serveur vers le client en lui spécifiant les différents paramètres à utiliser
(auparavant le serveur fait 2 pings pour savoir si l'adresse qu'il propose n'est pas déjà utilisée...)
-> *DHCPOFFER*
3. Client donne son accord sur cette adresse en faisant un broadcast
-> *DHCPREQUEST*
4. Le serveur acquiesce à cet accord en répondant en unicast
-> *DHCPACK*

Pour libérer une adresse IP, utilisation de *DHCPRELEASE*

Utilisable que sur le réseau LAN (par défaut)

Abandonné pour IPv6, et remplacé par ICMPv6

ICMP

✦ *ICMP : Internet Control Message Protocol*

- ✦ Permet le contrôle des erreurs de transmission
- ✦ Composé principalement :
 - D'un entête IP
 - D'un type de message
- ✦ Différent type de message :
 - Type 0 : echo réponse
 - réponse au type 8
 - Type 3 : destinataire inaccessible
 - Type 5 : redirection
 - Type 8 : demande echo
 - permet de savoir si une @IP répond (ping)

NTP - Généralité

✦ NTP – Network Time Protocol

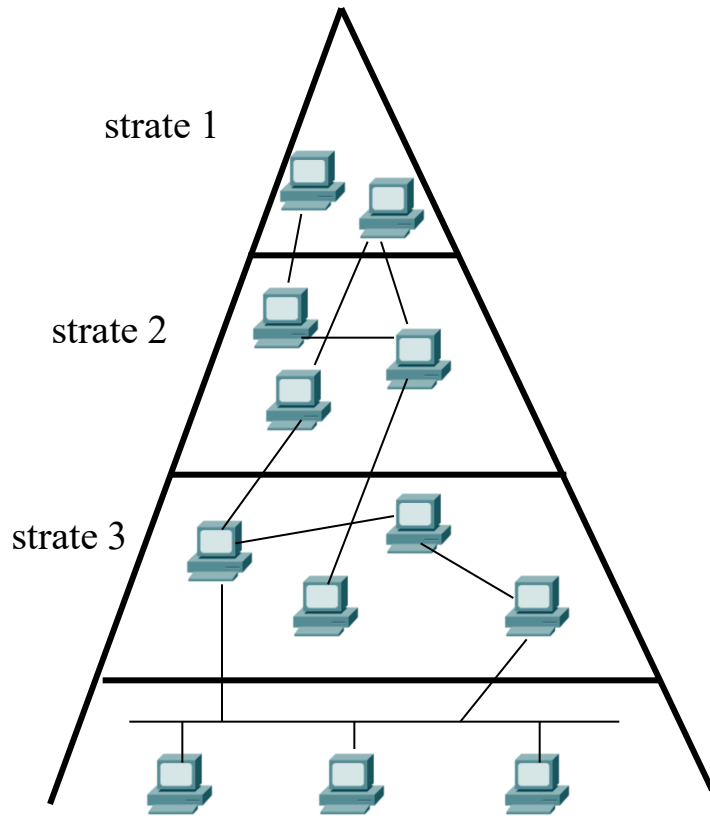
- ✦ Protocole réseau permettant de mettre à jour son horloge en se synchronisant sur des serveurs de temps présents sur Internet.

➔ **But : avoir un temps universel sur toutes les machines**
(compilation séparée, synchronisation des processus, etc...)

-Contraintes :

- la différence entre deux machines doit être inférieure à une certaine valeur
- l'horloge d'une machine avance continûment dans le temps

NTP – Principe



Utilisation de la notion de strate

- **strate 1** : serveurs qui sont synchronisés sur l'heure UTC
- **strate 2** : serveurs qui se synchronisent entre eux et avec la strate 1, serveurs publics en général.
- **strate 3** : serveurs qui se synchronisent entre eux et avec la strate 2, serveurs dans les entreprises.
- **strate 4** : ordinateurs de réseaux locaux qui se synchronisent sur la strate 3.

D'après la norme, 15 couches maximum !!! → 4 couches.

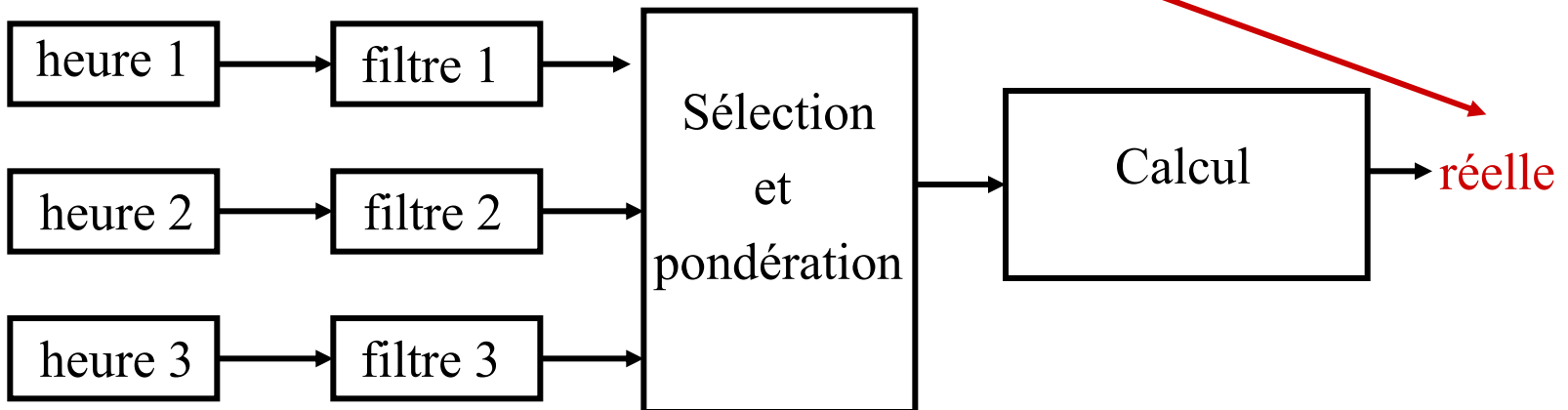
NTP – Synchronisation (1)

✦ Synchronisation

→ mise en place d'une variable d'ajustement

besoin : {
- heure actuelle de la machine
- heure réelle de la machine

- ◆ Calcul de **l'heure réelle** réalisé par le biais des différentes horloges récupérées



NTP –Synchronisation (2)

-
- ✦ Codage du temps sur 64 bits (Timestamp)
 - ✦ Date du début du codage : 1 janvier 1900
 - ✦ validité jusqu'en 2036...

 - ✦ Pour calculer l'heure transmise, il faut connaître le délai de propagation
Hypothèse : temps aller = temps retour

 - ✦ La trame contient différents champs dont :
 - la version du protocole
 - le mode d'échange : client/serveur, symétrique, multicast
 - 3 timestamps :
 - Originate timestamp
 - Receive timestamp
 - Transmit Timestamp

Quel calcul doit-on faire pour trouver le délai de transmission ?

Service de messagerie

- ✦ Autre nom : courrier électronique, e-mail, courriel
 - ➔ permet d'échanger des messages et des fichiers

- ✦ Il nécessite un serveur de messagerie accessible à partir d'internet. Le serveur dispose d'une boîte à lettre (BAL) pour chaque client géré par la messagerie.

- ✦ Les messages sont stockés par le serveur de messagerie, en attendant que le client vienne consulter sa boîte aux lettres.
 - ◆ le message peut être lu :
 - en *mode online* - message stocké sur le serveur et lu à distance
 - en *mode offline* – message déplacé sur la station client et effacé du serveur