

# Réseaux – couche transport

## UDP et TCP

IUT-2  
Département Informatique

25 avril 2023

# Sommaire

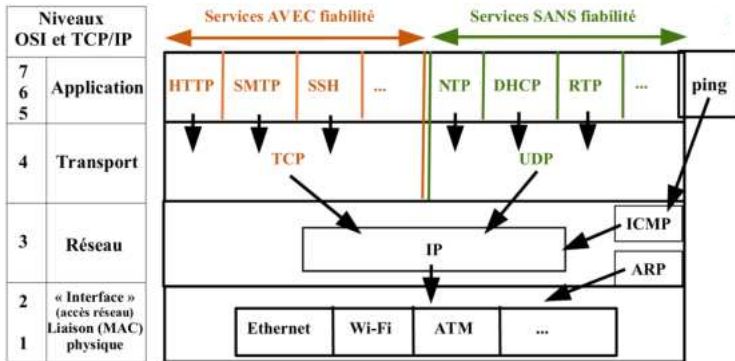
Le niveau transport

Protocole UDP

Protocole TCP

Glossaire46

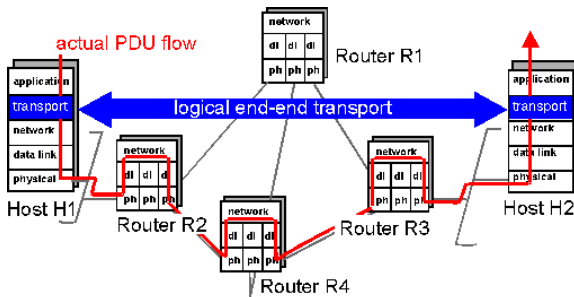
# Niveau transport dans l'architecture TCP/IP



- ▶ User Datagram Protocol (UDP) : service de transport « simple » sans connexion
- ▶ Transmission Control Protocol (TCP) : service de transport avec connexion (fiabilité des connexions)

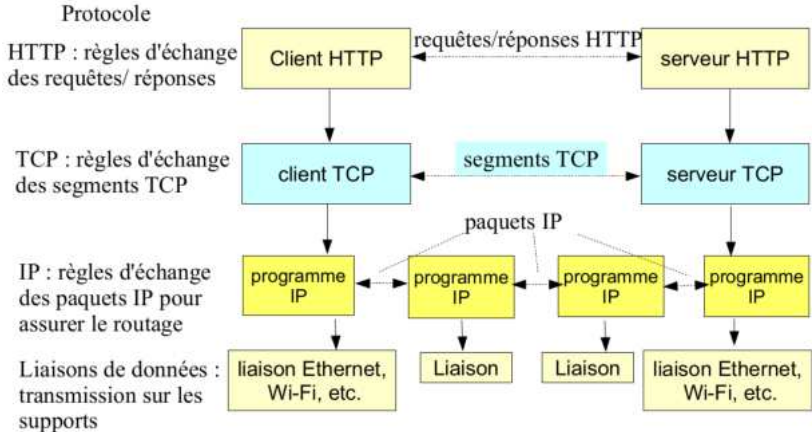
# Couche Réseau et Transport

- ▶ Couche réseau : communication logique entre stations.
- ▶ Couche transport : communication logique entre des processus (sur des machines différentes).
- ▶ Segmentation :
  - ▶ Émetteur : les messages applicatifs sont fragmentés en segments, puis passés à la couche réseau.
  - ▶ Récepteur : les segments sont ré-assemblés en messages applicatifs puis passés à la couche application.



d'après cahayainternettechnology

# Exemple des protocoles HTTP/TCP/IP



# Mise en place du niveau transport

Objectif :

Transport des datagrammes ou des flots d'octets entre un processus application client et un processus application serveur.

Identification des processus client et serveur :

- ▶ L'adresse IP ne suffit pas : plusieurs applications tournent sur une même station.
- ▶ Le numéro de processus système ne convient pas car le « PID » est local et dépend du système utilisé.
- ▶ **Numéro de port** : champ sur 16 bits de l'en-tête TCP ou UDP.

Modèle client/serveur :

- ▶ toute communication est initiée par un **client**.
- ▶ le numéro de port et l'adresse IP du serveur doivent être connus du (des) client(s) avant toute communication.
- ▶ les échanges de données sont ensuite symétriques.

# Ports

Valeur sur 16 bits qui identifie une connexion avec processus sur une station (client ou serveur).

De 0 à 1023 ce sont les ports système. Il faut avoir des privilèges admin pour les utiliser. Par exemple :

- ▶ 22 : pour SSH
- ▶ 25 : pour SMTP
- ▶ 80 : pour HTTP

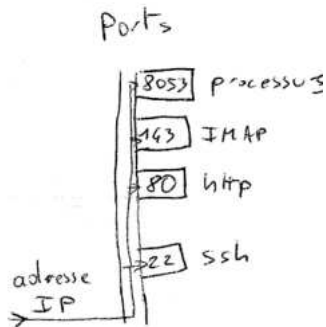
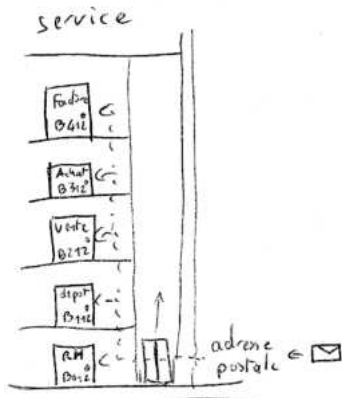
De 1024 à 49151 les ports enregistrés par l'Internet Assigned Numbers Authority (IANA), qui peuvent être utilisés sans privilèges admin.

- ▶ 5432 : postgresQL.
- ▶ 8080 : port HTTP alternatif ou tomcat.

De 49152 à 65535 ports dynamiquement alloués pour les processus.

On dit que les applications “écoutent” (*listen*) sur un port.

# Ports : comparaison avec le système postal





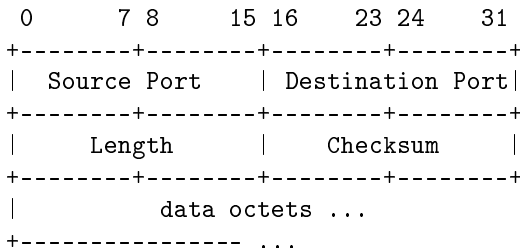
# Protocole UDP

Principe : service de transport minimum et simple [rfc, 1980]

Services offerts :

- ▶ Communication directe, dite « sans connexion »
- ▶ Encapsulation : dans un paquet IP (taille max = celle de IP).
- ▶ Pas de fragmentation/ré-assemblage (mais fait par IP).
- ▶ Multiplexage (plusieurs communications UDP en même temps) : possible en utilisant des n° de port différents.
- ▶ PAS de contrôle d'erreur de transmission ni de flux.

Format de l'en-tête d'un datagramme UDP :



# UDP propriétés

## UDP

- ▶ Transmission ou réception des paquets dans un ordre non contraint  
→ adapté aux applications en temps réel où la latence peut être un problème.

## Adapté pour

- ▶ Les protocoles basés sur les transactions, tels que Dynamic Host Configuration Protocol (DHCP) ou Network Time Protocol (NTP).
- ▶ Les cas où la correction d'erreurs en temps réel n'est pas nécessaire (jeux vidéo, visioconférence, streaming de contenu audio-visuel de loisir, VoIP).

# UDP Bilan

UDP 2 propriétés en plus d'IP.

- ① numéros de port pour distinguer les différentes connexion des utilisateurs.
  - ② Somme de contrôle sur une partie de l'en-tête IP + l'en-tête UDP + les données UDP, pour vérifier l'intégrité des données à l'arrivée.
- 
- ▶ Rapide et adapté au broadcast :
    - ▶ pas de retransmission,
    - ▶ pas d'ordonnancement des paquets.
  - ▶ Non fiable :
    - ▶ pas de vérification que le destinataire est prêt à recevoir,
    - ▶ pas de gestion des cas de duplications ou de perte de paquets,
    - ▶ aucune garantie que les octets soient tous reçus.

# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

Transmission TCP en full duplex

Ouverture, gestion des pertes et fermeture en TCP

Numéros de séquence

Contrôle de congestion

Bilan TCP

Glossaire46

# Protocole TCP

Principes [rfc, 1981] :

- ▶ couche logicielle qui masque aux applications les contraintes du réseau (hors les délais induits!).
- ▶ service de transport fiable en mode flot d'octets.

Notion de connexion TCP associée au couple constitué du processus client et du processus serveur :

- ▶  $\langle @IP_{\text{serveur}}, N_{\text{portserveur}}, @IP_{\text{client}}, N_{\text{portclient}} \rangle$

Réalisation du mode flot :

- ▶ Ouverture de connexion, préalable à tout échange de données.
- ▶ Numérotation des octets de données à émettre.
- ▶ Émission et réception des octets en blocs de segments, indépendamment des dépôts et retraits faits par les programmes applicatifs → stockage en mémoire (tampon) par TCP.

# Protocole TCP

Transfert fiable : livraison des segments dans l'ordre et gestion des segments perdus

Utilisation des numéros de séquence :

- ▶ pour réordonner le flux original

Utilisation des acquittements (ACKs) émis par le destinataire.

- ▶ Retransmission segments perdus

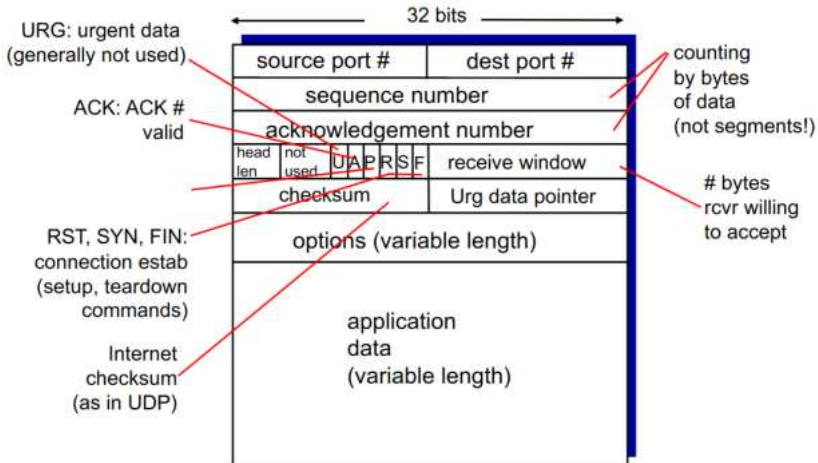
## Contrôle de flux

TCP fournit au destinataire un moyen de contrôler le débit des données envoyées par la source pour que l'émetteur n'émette pas trop vite par rapport au récepteur.

## Contrôle de congestion

- ▶ Pertes interprétées comme signal de congestion dans le réseaux.
- ▶ L'émetteur réagit en diminuant le débit de transmission.

# Format des paquets (segments) TCP



d'après Kurose & Ross

# Champs de l'en-tête TCP

**Sn : Sequence number** : n° du premier octet contenu dans le champ info (n° dans le flot de données d'émission).

**An : Acknowledge number** : n° du prochain octet attendu dans le flot de données reçues, acquitte les octets de n° inférieurs.

**AW : Advertised receive Window** : fenêtre de réception (notée ici W).

**H.lg : Header Length** : longueur de l'en-tête, en mots de 32 bits (si pas d'option, H.lg=5).

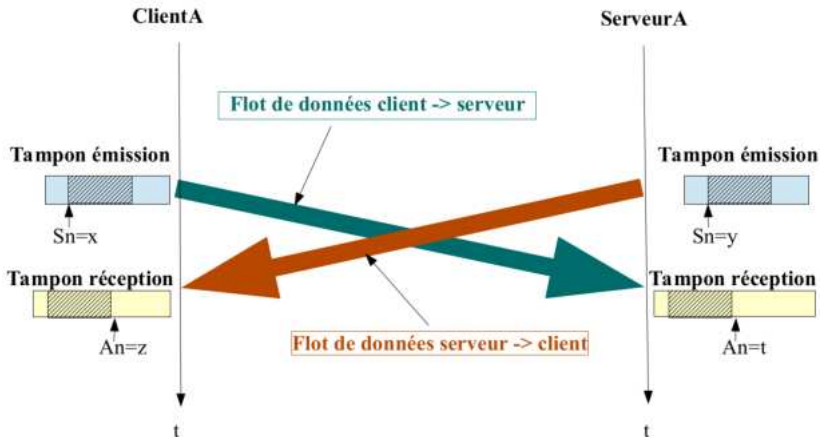
**Control (Flags)** : <0, 0, URG, ACK, PSH, RST, SYN, FIN>

- ▶ **SYN** : demande d'ouverture de connexion (O).
- ▶ **FIN** : demande de fermeture de connexion (F).
- ▶ **ACK** : le paquet acquitte des données ou une demande O/F.
- ▶ **PSH** : le segment contient des données qui peuvent être délivrées à l'application.

**Checksum** : même calcul que pour l'UDP.



# Représentation d'une connexion TCP et des tampons nécessaires



# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

Transmission TCP en full duplex

Ouverture, gestion des pertes et fermeture en TCP

Numéros de séquence

Contrôle de congestion

Bilan TCP

Glossaire46

# Étapes de la transmission de données

- ① L'application dépose ses données dans le tampon émission.
- ② Le programme TCP émet ces données en blocs appelés « segments ».
- ③ Le programme TCP côté réception :
  - 3.1 Récupère et stocke ces segments dans le tampon réception.
  - 3.2 Envoie à l'émetteur un acquittement à chaque segment reçu.
- ④ Le programme application récupère les données dans le tampon réception.
- ⑤ Le programme TCP côté émission libère l'espace du tampon émission des données acquittées par le récepteur.
- ⑥ Le programme TCP côté réception libère l'espace du tampon réception des données récupérées par l'application.

# Émission en TCP-1

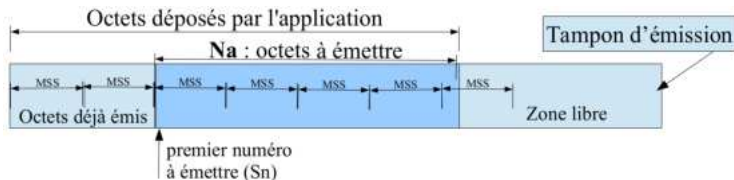
- ▶ L'application dépose ses données dans le tampon émission : c'est le « flot » d'émission, l'ordre dans le tampon sera celui de la transmission.
  - ▶ TCP envoie ce flot par segments : il découpe les données du tampon (flot) en segments.
  - ▶ TCP émet des octets du flot dès qu'il le peut : s'il y a des octets non encore émis et si la connexion le lui permet.
- La formule qui donne la taille de chaque segment émis est :

$$\text{MIN}(Na, \text{MSS}, W)$$

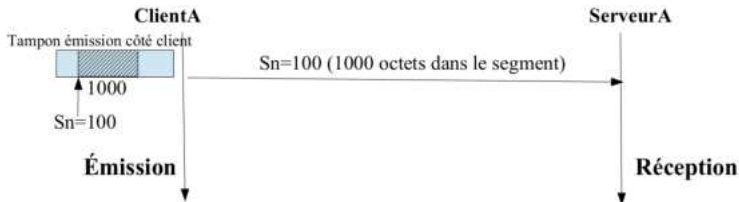
- ▶ **Na** : Nb d'octets « application » à émettre, stockés dans le tampon d'émission.
- ▶ **Maximum Segment Size (MSS)** : dépend de la taille max de la trame Maximum Transfer Unit (MTU).  
**MSS = MTU – en-têtes IP et TCP.**
- ▶ **W** : fenêtre = volume autorisé par le programme TCP réception.

# Émission en TCP-2

Représentation du tampon d'émission (flot de données à émettre) :

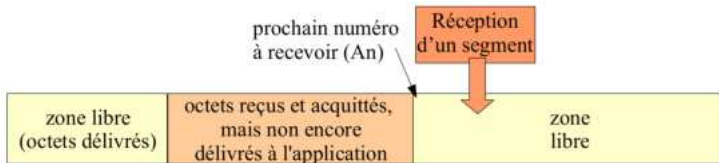


Exemple d'émission de données (MSS = 1000)

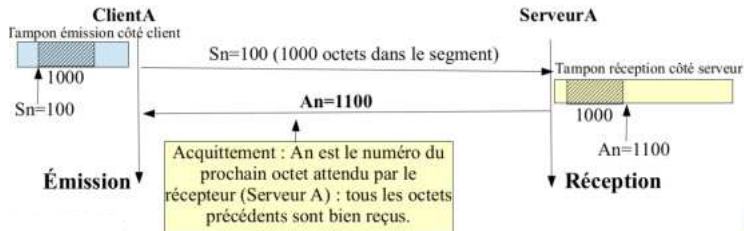


# Émission en TCP-2

Stockage des données reçues dans un tampon réception :



Acquittement dès réception : le champ **An** indique toujours le n° du prochain octet à recevoir (= les n° précédents sont reçus sans erreur) :



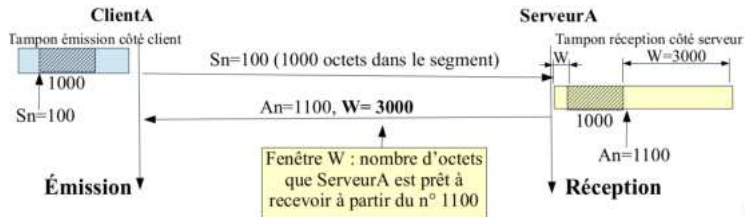
# Fiabilisation des échanges en TCP

Contrôle d'erreur : numérotation des octets, acquittement des octets reçus et utilisation du délai de réémission **T<sub>max</sub>** pour réémettre les données non acquittées.

- ▶ **S<sub>n</sub>** : numéro du premier octet émis dans le segment.
- ▶ **A<sub>n</sub>** : prochain numéro d'octet attendu (tous les octets de numéro inférieur sont correctement reçus).

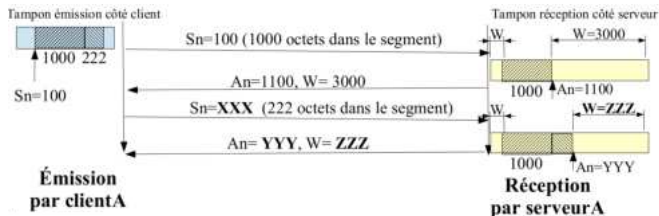
Conservation de l'ordre des octets déposés par l'application.

Contrôle de flux : le récepteur envoie **W** (= place disponible à la réception).



# Exemple d'échange de données

Dans le scénario ci-dessous, quelles sont les valeurs de **XXX**, **YYY** et **ZZZ** ?



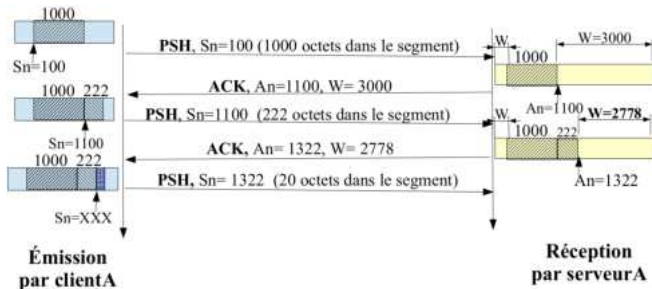


# Champ « Control » et fenêtre **W**

Control (Flags) :  $\langle 0, 0, \text{URG}, \text{ACK}, \text{PSH}, \text{RST}, \text{SYN}, \text{FIN} \rangle$

- ▶ **ACK** : le paquet acquitte des données (indiqué par la valeur  $A_n$ ).
- ▶ **PSH** : le segment contient des données qui peuvent être délivrées à l'application.

Fenêtre **W** : le récepteur indique combien d'octets il peut encore recevoir.



# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

**Transmission TCP en full duplex**

Ouverture, gestion des pertes et fermeture en TCP

Numéros de séquence

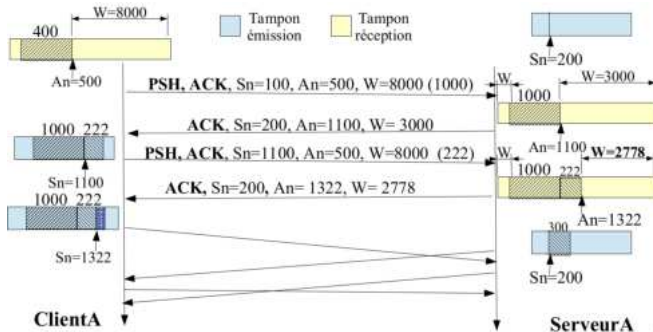
Contrôle de congestion

Bilan TCP

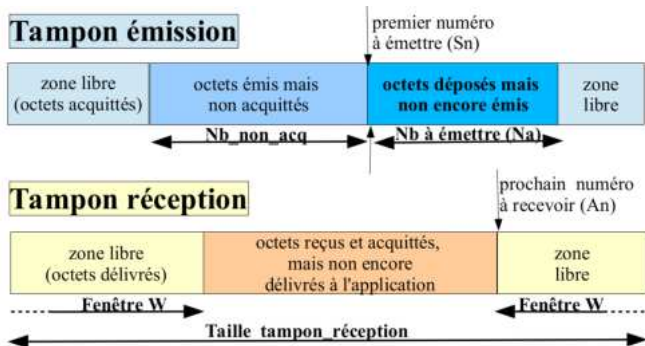
Glossaire46

# Simultanéité des échanges de données

- ▶ Les données sont émises et reçues dans les 2 sens de la connexion et les segments peuvent se « croiser » dans l'(inter)-réseau.
- ▶ Les drapeaux et numéros **Sn**, **An**, **W** sont dans chaque segment !



# État des tampons émission et réception



Quelle est la valeur de **W** à émettre par chaque programme TCP ?

C'est la zone libre de son tampon réception :

$W = Taille\_tampon\_reception - octets\ reçus\ et\ non\ délivrés.$

Exemple : tampon réception de 10000 octets dont 3000 occupés

→  $W = 7000$

# Rôle des tampons TCP

Chaque programme TCP a deux tampons (buffer) :

- ▶ Un à l'émission
- ▶ Un à la réception

Cela permet de désynchroniser le dépôt des données et l'envoi d'une part et la réception et la délivrance des données d'autre part.

→ Permet la mise en place des contrôles d'erreur et de flux.

# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

Transmission TCP en full duplex

Ouverture, gestion des pertes et fermeture en TCP

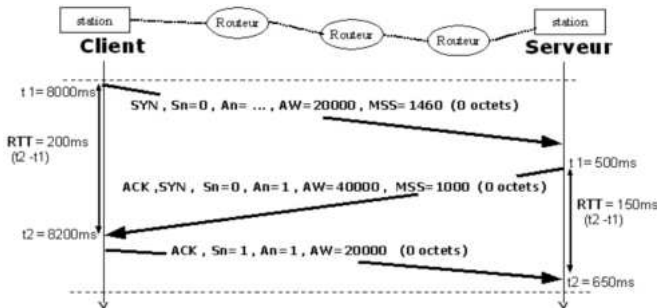
Numéros de séquence

Contrôle de congestion

Bilan TCP

Glossaire46

# Échanges à l'ouverture d'une connexion



Initialisation de  **$S_n$ ,  $A_n$ ,  $AW$**  (taille du tampon réception) de chaque côté de la connexion.

Valeur de  **$MSS$**  négociée : minimum des valeurs échangées.

Initialisation du délai de réémission  $T_{max} = 2 * RTT$ ,  
avec Round-Trip Time (RTT) (délai d'aller-retour sur la connexion).

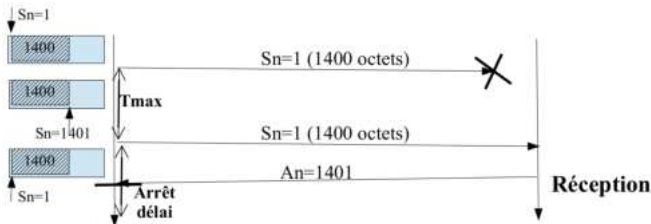
# Échange avec perte de données

Détection d'erreur : si l'acquittement de données n'arrive pas au bout du délai **T<sub>max</sub>**, les données sont considérées comme perdues.

Correction d'erreur : les données sont réémises.

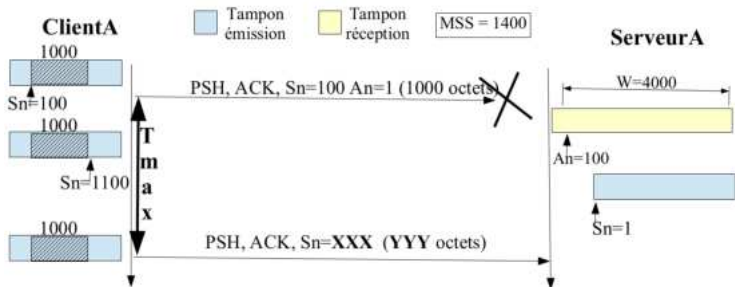
Mise en œuvre :

- ▶ Tampon émission, **S<sub>n</sub>**, **A<sub>n</sub>**.
- ▶ Délai de garde initialisé à chaque segment de données émis.
- ▶ En réception, tout segment reçu est aussitôt acquitté.



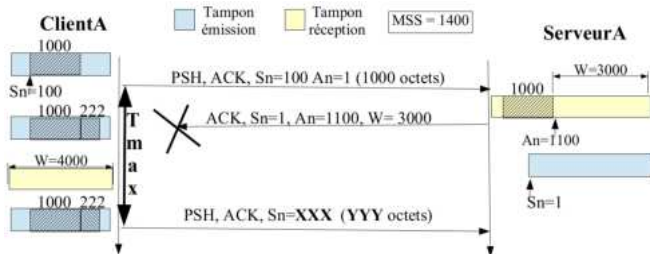


# Échange avec perte de données : exercice 1



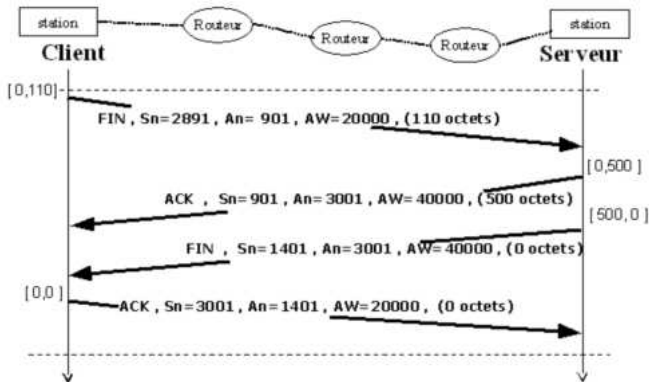
Quelles sont les valeurs de **XXX** et **YYY** du segment ré-émis à la fin du délai **T<sub>max</sub>** ?

## Échange avec perte de données : exercice 2



Quelles sont les valeurs de **XXX** et **YYY** du segment ré-émis à la fin du délai **Tmax** ?

# Échanges à la fermeture d'une connexion



Après une demande **FIN**, les tampons sont vidés, puis la connexion est arrêtée.

De nombreux cas sont à considérer : double envoi de FIN, etc.

# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

Transmission TCP en full duplex

Ouverture, gestion des pertes et fermeture en TCP

**Numéros de séquence**

Contrôle de congestion

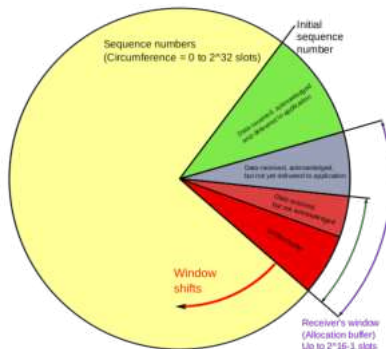
Bilan TCP

Glossaire46

# Évolution des numéros de séquence et ACK

- ▶  $S_n$  indique le numéro du premier octet du segment transmis
- ▶  $S_n$  est un nombre compris entre 0 et  $2^{32}$
- ▶  $S_n$  est implanté comme un entier non signé
- ▶  $S_{n+1} = (S_n + len) \bmod 2^{32}$

ou mod est l'opérateur modulo



(source wikipedia)

## Évolution

- ▶ Si le drapeau SYN est présent, l'émetteur devrait recevoir  $ACK = (S_n + 1) \bmod 2^{32}$
- ▶ Si le drapeau SYN n'est pas présent, l'émetteur devrait recevoir  $ACK = (S_n + len) \bmod 2^{32}$

# Numéro de séquence initial

Problème : plusieurs connexions TCP ayant les mêmes sources et destinations peuvent co-exister (redémarrage, demi-fermeture, etc.)

- ▶ les numéros de séquence peuvent se mélanger !
- choix pseudo aléatoire du numéro de séquence initial Initial Sequence Number (ISN)

$ISN = M + F(localip, localport, remoteip, remoteport, secretkey)$

**M** est une horloge et **F()** un générateur pseudo aléatoire prenant en paramètre les propriétés de la connexion et une clé secrète.

Le numéro du premier octet est toujours : **ISN+1**.

# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

Transmission TCP en full duplex

Ouverture, gestion des pertes et fermeture en TCP

Numéros de séquence

Contrôle de congestion

Bilan TCP

Glossaire46

# Congestion

La congestion d'un réseau informatique est la condition dans laquelle une augmentation du trafic (flux) provoque un ralentissement global de celui-ci (délai, perte de paquet, impossibilité d'établir des connexions).

TCP utilise quatre algorithmes : slow start, congestion avoidance, fast retransmit, et fast recovery définis dans la RFC 5681 [Blanton et coll., 2009].

Ces algorithmes assurent :

- ▶ une adaptation du flux de chaque connexion TCP à un niveau évitant la congestion générale.
- ▶ Une répartition équivalente des débits de chaque connexion

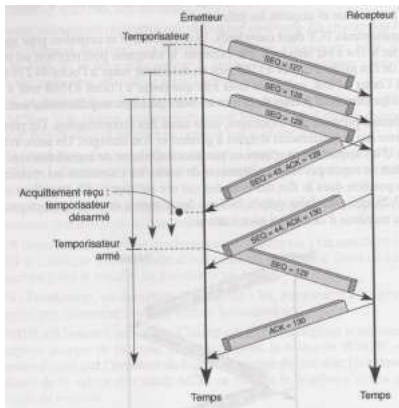


# Fenêtre de congestion et retransmission

Une fenêtre de congestion (CW) indique le nombre  $N$  de segments de taille MSS qui peuvent être envoyés en une seule fois.

Il y a donc deux fenêtres dans une connexion TCP (AW et CW) et le minimum des deux détermine la quantité de données envoyée.

Un ACK peut acquitter plusieurs segments. Si un ACK n'arrive pas au bout d'un certain temps, le segment est retransmis.



(source [Pujolle, 2018])

# Slow start and congestion avoidance

Lors de la détection d'une congestion, la connexion redémarre lentement puis la taille  $N$  de la fenêtre de congestion croît progressivement.

## slow start

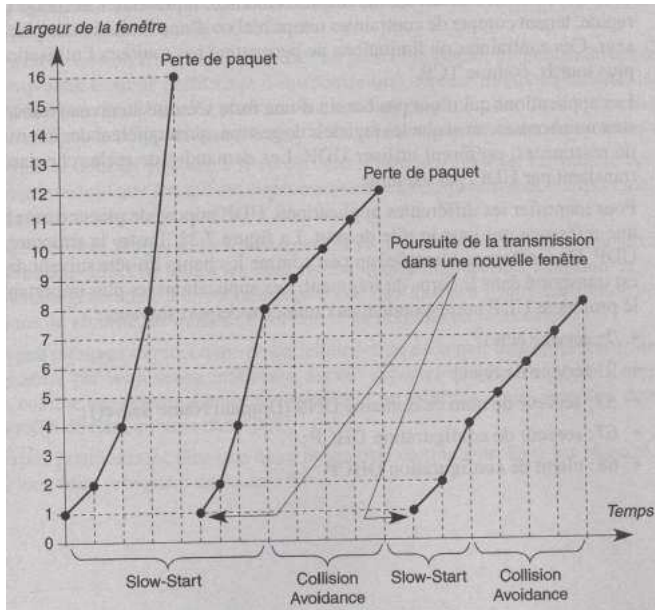
$N = 1$  puis double jusqu'à atteindre un seuil (*ssthresh*)

## congestion avoidance

$N = N+1$  jusqu'à une congestion

dans ce cas on redémarre avec un seuil (*ssthresh*) égale à  $N/2$

# Slow start and congestion avoidance



# Sommaire

Le niveau transport

Protocole UDP

Protocole TCP

Le Protocole TCP

Transmission en TCP

Transmission TCP en full duplex

Ouverture, gestion des pertes et fermeture en TCP

Numéros de séquence

Contrôle de congestion

Bilan TCP

Glossaire46

# Protocole TCP

Mode connecté :

- ▶ Transfert des segments dans des paquets IP
- ▶ Transmission dans les deux sens (full-duplex)
- ▶ Pas de détection de fin de connexion

Transfert fiable : livraison des segments dans l'ordre et gestion des segments perdus

Utilisation des numéros de séquence et des acquittements (ACKs) :

- ▶ Réordonner le flux original
- ▶ Retransmission des segments perdus

Contrôle décentralisé :

- ▶ Contrôle de flux : contrôle du débit pour une connexion par rapport au récepteur
- ▶ Contrôle de congestion : contrôle du débit pour une connexion par rapport à l'état du trafic sur le réseau

# Glossaire I

**DHCP** Dynamic Host Configuration Protocol. 10

**IANA** Internet Assigned Numbers Authority. 7

**ISN** Initial Sequence Number. 38

**MSS** Maximum Segment Size. 20

**MTU** Maximum Transfer Unit. 20

**NTP** Network Time Protocol. 10

**RTT** Round-Trip Time. 31

**TCP** Transmission Control Protocol. 3

**UDP** User Datagram Protocol. 3

# Références |



(1980).

User Datagram Protocol.

RFC 768.



(1981).

Transmission Control Protocol.

RFC 793.



Blanton, E., Paxson, D. V. et Allman, M. (2009).

TCP Congestion Control.

RFC 5681.



Pujolle, G. (2018).

*Les réseaux.*

Editions Eyrolles, 9 édition.