# Rappels Chapitre 3 et 4 pour laboratoire 4

☐ Le contenu de cette présentation est basé sur le livre de Kurose et Ross et de la documentation y jointe :

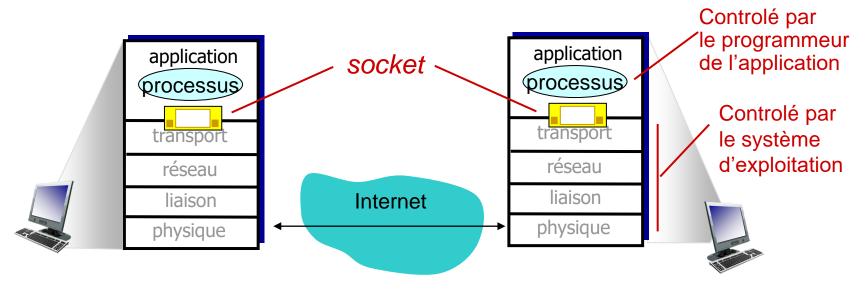
Computer Networking: A Top Down Approach, 6ème édition. Jim Kurose, Keith Ross Addison-Wesley, Mars 2012,

ISBN-13: 978-0132856201

Computer Networking

#### Interface de connexion (Sockets)

- □ Processus émet/reçoit des messages à travers son socket
- Socket analogue à une porte
  - O Processus d'émission met les messages à l'extérieur
  - Processus d'émission compte sur l'infrastructure transport de l'autre côté de la porte pour transférer les messages à la porte du côté réception



Socket: Interface entre un processus de la couche application et un protocole de la couche transport. Permet de différencier les données sortant d'un processus ou allant vers un processus.

#### structure d'un segment TCP

URG: données urgentes (généralement, non utilisé)

ACK: # ACK valide

PSH: pousse les données maintenant (pas très utilisé) RST, SYN, FIN: étab. de connexion (commandes d'étab. et de libération)

somme de contrôle (comme pour UDP)

source port # dest port #

sequence number

acknowledgement number

head not UAPRSF Receive window

cheeksum Urg data pnter

Options (taille variable)

32 bits

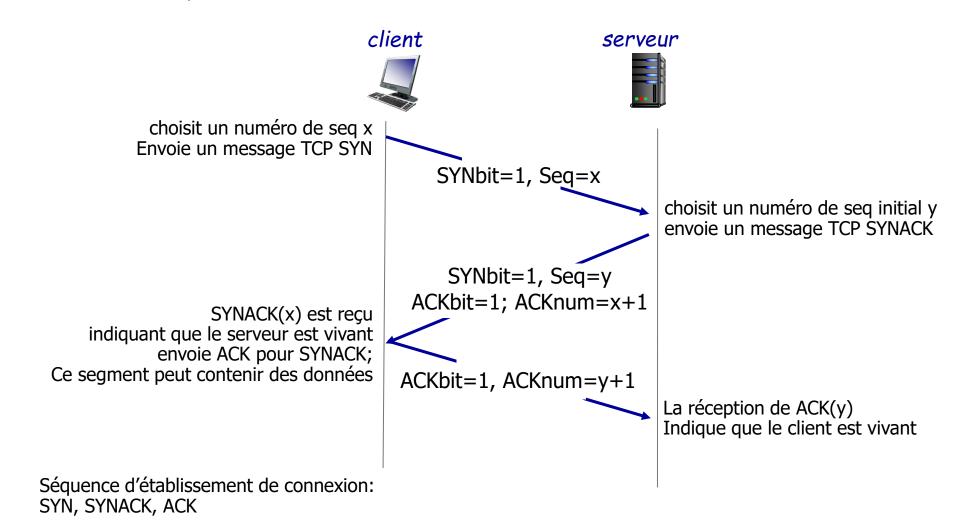
Données de l'application (taille variable)

compter les données en octets (pas en segments!)

> Nombre d'octets que le récepteur peut accepter

Ex.: MSS, Timestamp, ...

## Ouverture d'une connexion TCP TCP 3-way handshake



### # séq. et ACKs dans TCP

#### Numéro de séquence:

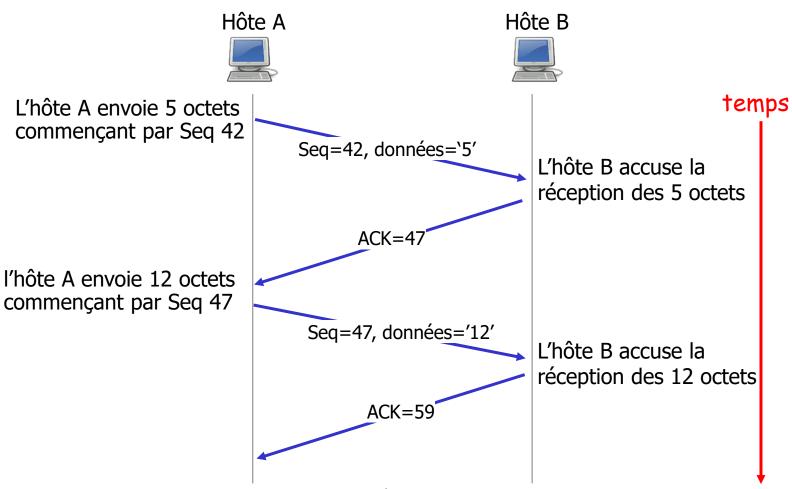
 le numéro du premier octet dans les données du segment

#### ACKs: Acquittement

- # de séq du prochain octet attendu
- ACK cumulatif

Q: comment le récepteur traite les segments non ordonnés?

 TCP ne le précise pas : cela dépendra de l'implémentation



Scénario simple: seul l'hôte A envoie des données

### # séq. et ACKs dans TCP

#### Numéro de séquence:

 le numéro du premier octet dans les données du segment

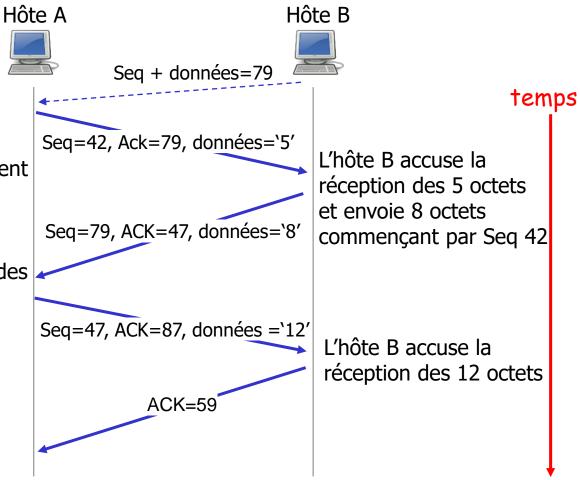
#### ACKs: Acquittement

- # de séq du prochain octet attendu
- ACK cumulatif

L'hôte A envoie 5 octets commençant par Seq 42 et accuse la réception des données reçues précédemment

l'hôte A accuse la réception des 8 octets et envoie 12 octets commençant par Seq 47

- Q: comment le récepteur traite les segments non ordonnés?
  - TCP ne le précise pas : cela dépendra de l'implémentation



Scénario simple: les deux hôtes envoient des données

### Format d'un datagramme IP

version du protocole IP

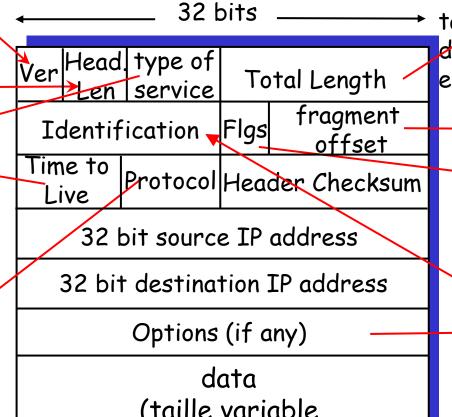
taille de l'entête (octets)-"type" de données-

nombre de sauts restants (décrémenté à chaque routeur)

les protocoles des couches supérieures

#### Taille de l'entête avec le TCP?

- 20 octets de TCP
- 🗖 20 octets de IP
- = 40 octets + données app



(taille variable,
Typiquement un segment TCP
ou UDP)

taille totale du dátagramme (données + entête) (octets)

Flags 3 bits.

DF MF

| | | |

MF= More Fragment

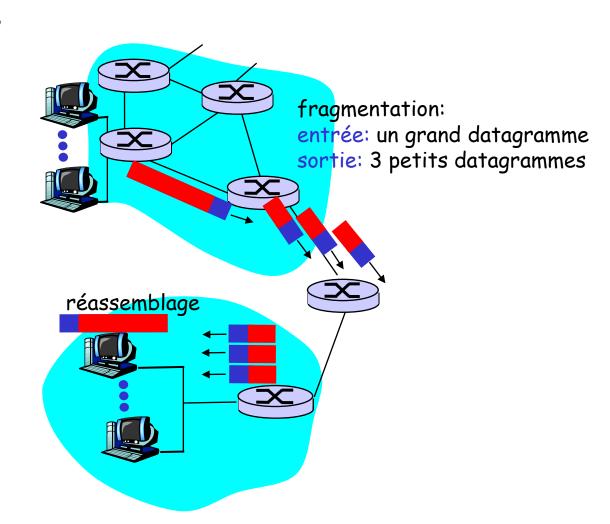
DF= Don't fragment

pour fragmentation/ Réassemblage

Par ex.
timestamp,
enregistrer la
Route (spécifier
la liste des
routeurs à visiter)

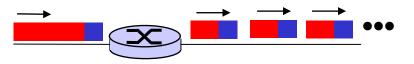
### Fragmentation IP & Réassemblage

- Un lien réseau a un MTU (maximal transfer unit) - taille de trame maximale.
  - différents types de liens, différents MTUs
- Un datagramme IP large est divisé ("fragmenté") dans le réseau
  - le datagramme est divisé en plusieurs datagrammes
  - "réassemblage" à la dest. finale seulement
  - les bits d'entête IP sont utilisés pour identifier les fragments



### IP Fragmentation & Réassemblage, exemple

- □ Datagramme de 420 octets (données + entête)
- MTU lien suivant = 128 octets
- Taille des entêtes 20 octets



length	ID	MF	offset
=420	=7	=0	=0

Donnez les valeurs des champs length, ID, MF, offset des fragments.

Règle: La taille des données dans les fragments doit être un multiple de 8 octets sauf pour le dernier fragment.

Taille des données à transmettre  $t_{data}$ 

$$t_{data}$$
 = 420-20 = 400 octets

Taille maximale des données dans un fragment  $t_{maxdatafrag}$ 

128-20 = 108 | 108/8 = 13,5 | 
$$t_{mf}$$
 = 13 x 8 = 104

Nombre de fragments  $n_{frag}$ 

400/104 = 3,8 Arrondi à l'entier supérieur  $\Rightarrow n_f = 4$ 

length	ID	MF	offset
124	7	1	0

124 7	1	13
-------	---	----

124 7	1	26
-------	---	----

108 7 0
---------