Chapitre 4:

Les protocoles TCP et UDP

Rappel

7	**Application**	ex. HTTP, HTTPS, Gopher, SMTP, SNMP, FTP, Telnet, NFS
6	**Présentation**	ex. ASCII, Unicode, MIME, XDR, ASN.1, SMB, AFP
5	**Session**	ex. ISO 8327 / CCITT X.225, RPC, Netbios, ASP
4	**Transport**	ex. TCP, UDP, SCTP, SPX, ATP
3	**Réseau**	ex. IP (IPv4 ou IPv6), ICMP, IGMP, X.25, CLNP, ARP, RARP, OSPF, RIP, IPX, DDP
2	**Liaison**	ex. Ethernet, Token Ring, PPP, HDLC, Frame relay, RNIS (ISDN), ATM, Wi-Fi, Bluetooth, ZigBee, irDA (Infrared Data Association)
1	**Physique**	ex. techniques de codage du signal (électronique, radio, laser, ...) pour la transmission des informations sur les réseaux physiques (réseaux filaires, optiques, radioélectriques ...)

UDP (User Datagramme Protocol)

- **RFC 768 : (** Request for Comments)

- UDP Fournit juste les fonctions de base pour la transmission, sans aucune garantie.

- UDP n'est jamais utilisé pour envoyer des données importantes comme les pages web, les informations de bases de données, etc..

- Les flux multimédias comme la video et audio, utilisent UDP car il offre la vitesse.

Introduction

La couche transport (couche 4) d'Internet dispose de deux protocoles pour la communication entre applications :

- **UDP (User Datagramme Protocol)** :
  protocole en mode **sans connexion**

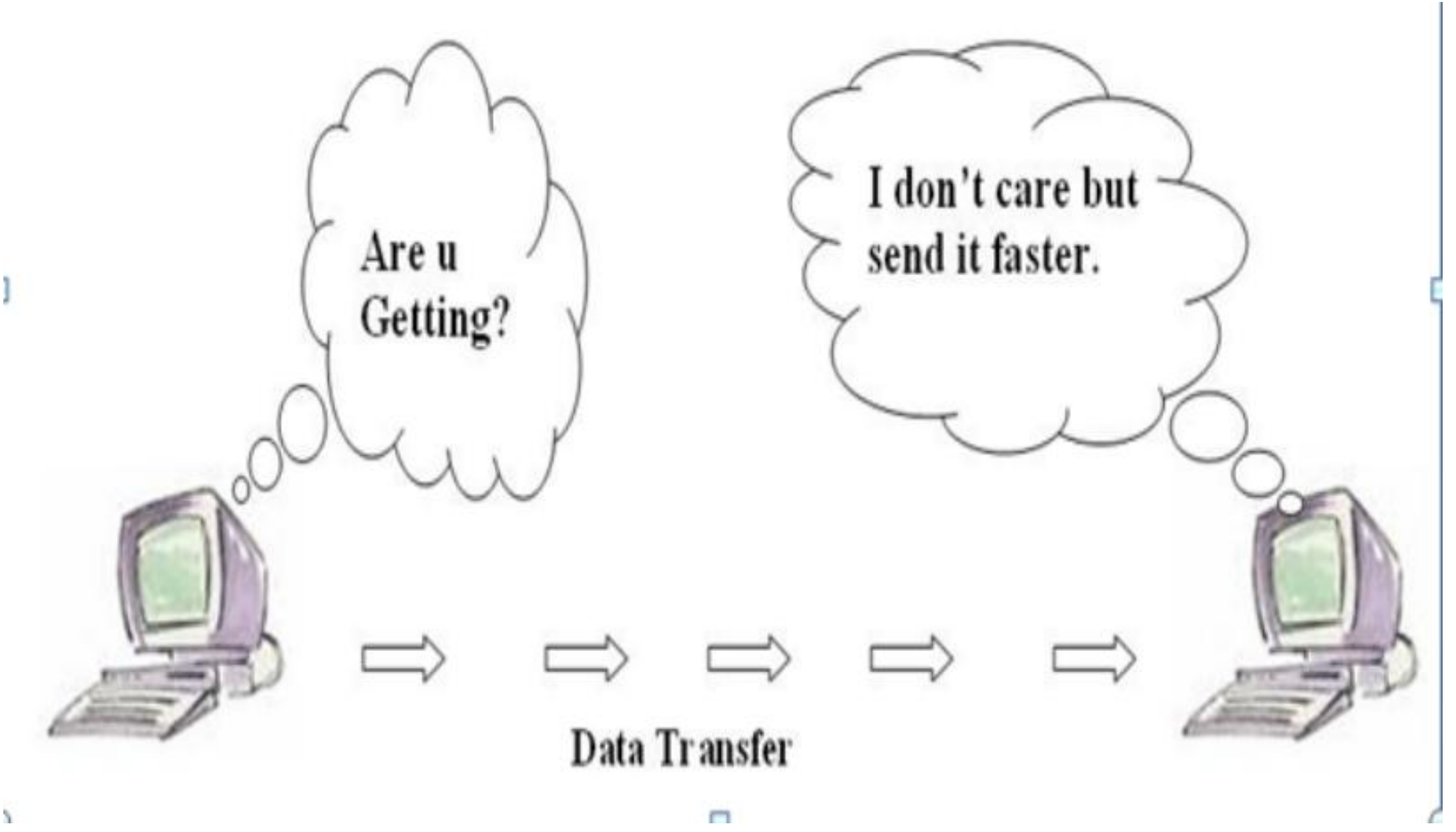
- **TCP (Transmission Control Protocol)** :
  protocole en mode orienté **connexion**

UDP : User Datagramme Protocol

UDP (User Datagramme Protocol)

- Service en mode non connecté
- Ordonnancement et arrivée des messages non garanti

UDP (User Datagramme Protocol)



UDP (User Datagramme Protocol)

Pas de vérification d'erreurs
Pas de correction d'erreurs

Téléphonie IP (VoIP)

Vidéo en continu

Pas de reconstitution ordonnée des données
Les données sont reconstituées selon l'ordre de réception.

Acheminement non fiable
Les segments perdus ne sont pas renvoyés.

Sans connexion
Pas d'établissement de session.

Pas de contrôle de flux
Pas de gestion de l'encombrement.

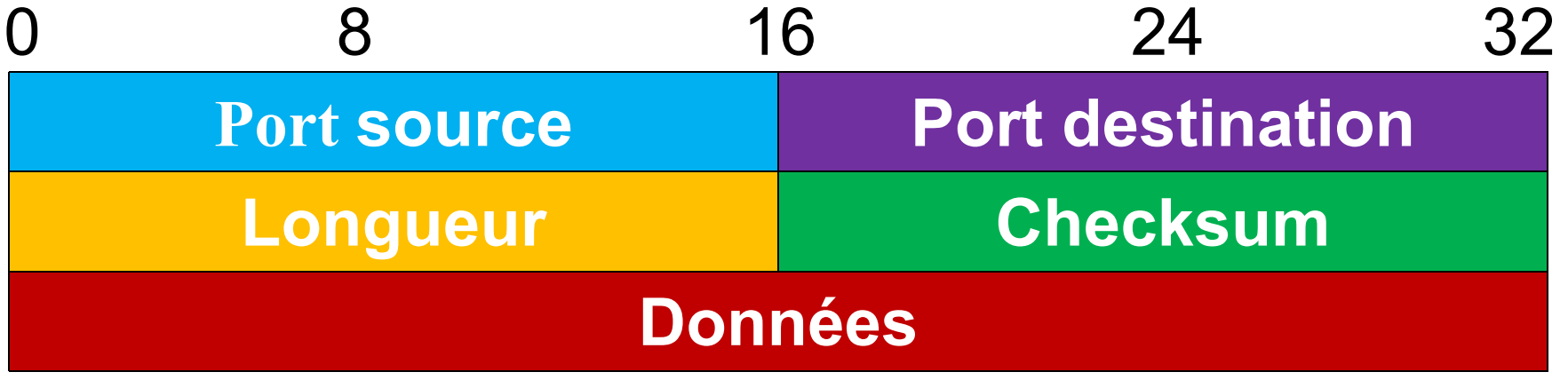
UDP : utilité des ports

plusieurs applications sont exécutés simultanément sur une même machine, on attribue à chacune une adresse unique codée sur 16 bits appelé : **un port**

- **L'adresse IP** sert donc à identifier de façon unique un ordinateur sur le réseau

- **le N° de port** indique l'application à laquelle les données sont destinées.

Datagrammes UDP



Port Source

- **depuis quel port le paquet a été envoyé.**

Port de Destination

- **à quel port le paquet doit être envoyé.**

Longueur

- **longueur totale (Octets) du segment UDP (en-tête+données). La longueur minimale est donc de 8 Octets (taille de l'en-tête).**

Somme de contrôle

- **assurer l'intégrité du paquet reçu.**

Elle est calculée sur l'ensemble de l'en-tête UDP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)

Classement des ports

Une assignation standard a été mise au point par l'**IANA** (***Internet Assigned Numbers Authority***), afin d'aider à la configuration des réseaux.

`1-1023` : services réservés s'exécutant avec des droits privilégiés (*root*)

`1024-49151` : services enregistrés auprès de l'IANA et pouvant s'exécuter avec des droits ordinaires

`49152-65535` : libres de toutes contraintes

Numéros de ports

. Les derniers numéros de ports peuvent être obtenus sur le site de IANA

. Sous Linux le fichier

/etc/services

contient les numéros de ports et les services associés.

Les ports standards

Certains ports sont réservés

N° port	Mot-clé	Description
7	ECHO	Echo
11	USERS	Active Users
13	DAYTIME	Daytime
37	TIME	Time
42	NAMESERVER	Host Name Server
53	DOMAIN	Domain Name Server
67	BOOTPS	Boot protocol server
68	BOOTPC	Boot protocol client
69	TFTP	Trivial File Transfer protocol
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management prot.

D'autres numéros de port (non réservés) peuvent être alloués dynamiquement aux applications.

TCP : Transport Control Protocol

Plan de la partie

- Introduction
- Structure d'un Segment TCP
- Ouverture et Clôture de connexion
- Mécanismes de fenêtre d'anticipation

Introduction

- **RFC 793**
- S'appuie sur IP (réseau non fiable)
- Communication en mode **connecté**
  1. Ouverture d'un canal
  2. Communication Full-Duplex
  3. Fermeture du canal

- TCP doit :
  - assurer la délivrance en séquence (**Arrivée** et **Ordre garanties**),
  - contrôler la **validité** des données reçues,
  - organiser les **reprises** sur erreur,
  - réaliser le **contrôle de flux :** régulation de la quantité de données transmises.

Segment TCP

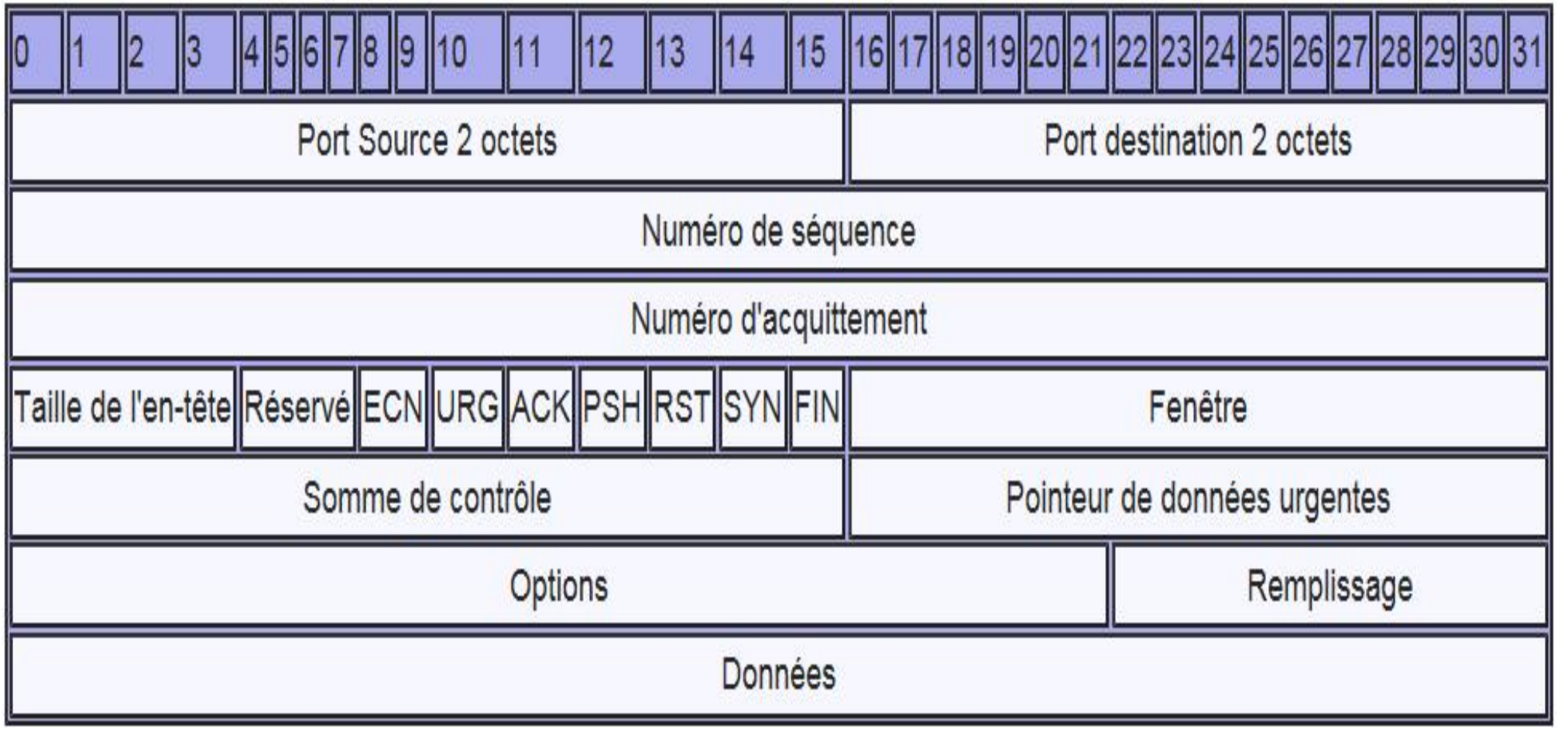
Notion de segment

Segment : Unité de transfert du protocole TCP

Types de Segments:

- Ouverture de connexions
- Transfert de données et acquittements
- Fermeture de la connexions

Format du segment TCP



Format du segment TCP

Port Source 2 octets	Port Destination 2 octets
Numéro de séquence	
Numéro d'acquittement	

- Port source : numéro du port source
- Port destination : numéro du port destination
- Numéro de séquence : numéro de séquence du premier octet de ce segment
- Numéro d'acquittement : numéro de séquence du prochain octet attendu
- Taille de l'en-tête : longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête)

Format du segment TCP

- **Drapeaux**
  - Réservé : réservé pour un usage futur
  - ECN : signale la présence de congestion,
  - URG : Signale la présence de données **urg**entes
  - ACK : signale que le paquet est un accusé de réception (**ack**nowledgement)
  - PSH : données à envoyer tout de suite (**pu**sh)
  - RST : rupture anormale de la connexion (**re**se**t**)
  - SYN : demande de synchronisation (SYN) ou établissement de connexion
  - FIN : demande la FIN de la connexion
- **Fenêtre** : taille de fenêtre demandée, c'est-à-dire le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception

Format du segment TCP

Somme de contrôle	Pointeur de données urgentes

les sommes de contrôle sur 16 bits, permettent la détection d'erreurs.

1. calculée par l'émetteur,
2. le destinataire recalcule la somme de contrôle du segment reçu,
3. si elle correspond à la somme de contrôle reçue,

le segment a été reçu sans erreur.

Format du segment TCP

- **Pointeur de données urgentes** : position relative des dernières données urgentes

- **Options** : facultatives

Format du segment TCP

Options	Remplissage
Données	

- **Remplissage** : zéros ajoutés pour aligner les champs suivants du paquet sur 32 bits, si nécessaire

- **Données** : séquences d'octets transmis par l'application

Ouverture et Clôture de connexion

Connexion

Une connexion de type circuit virtuel est établie :

Une connexion

=

une paire d'extrémités de connexion

Et

Une extrémité de connexion

=

couple
(@ IP , N°port)

Exemple de Connexion

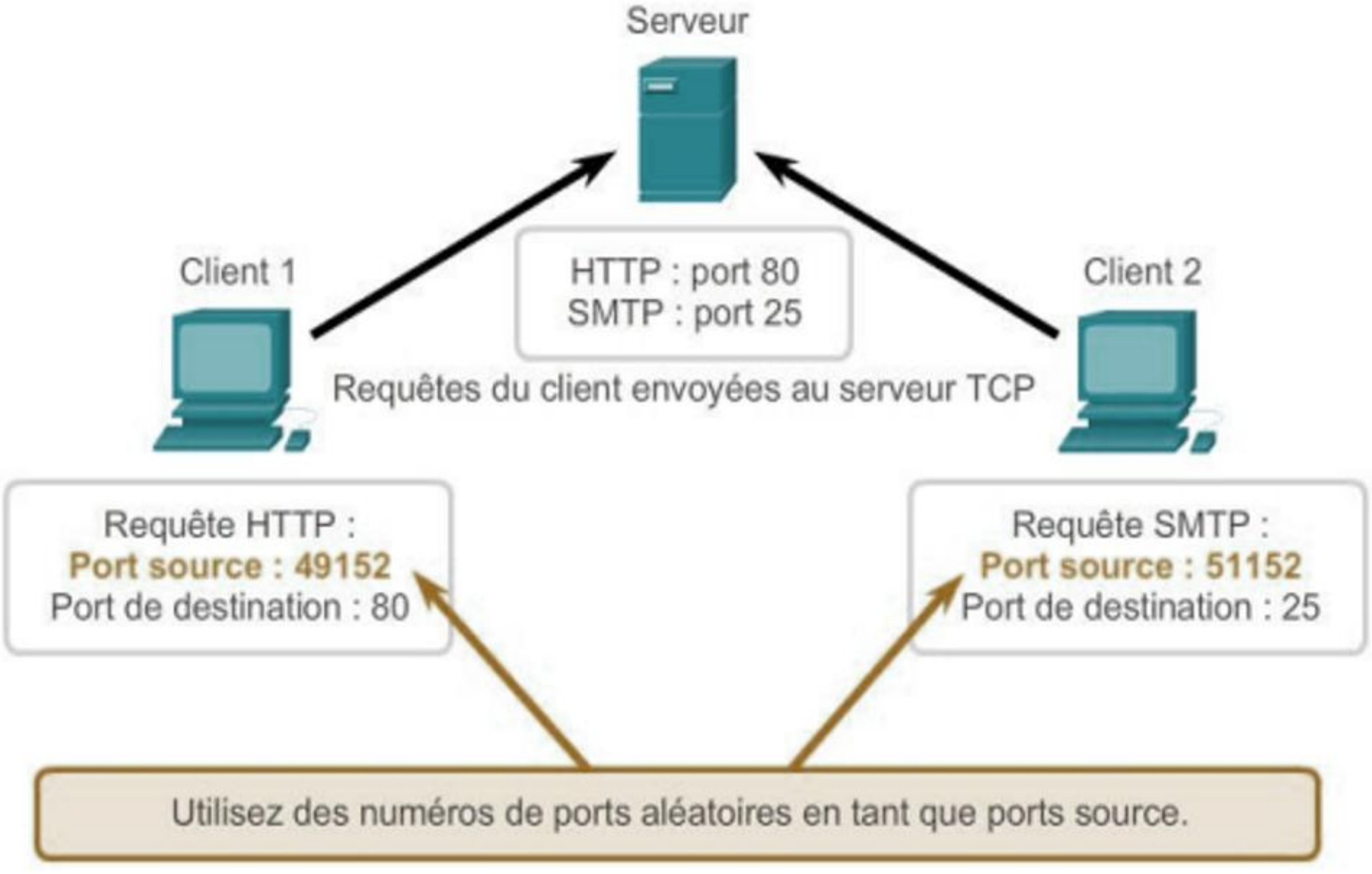
(@IP source, Port source) (@IP dest, Port dest)

- ✓ Une extrémité de connexion peut être partagée par plusieurs autres extrémités de connexions (multi-instanciation)

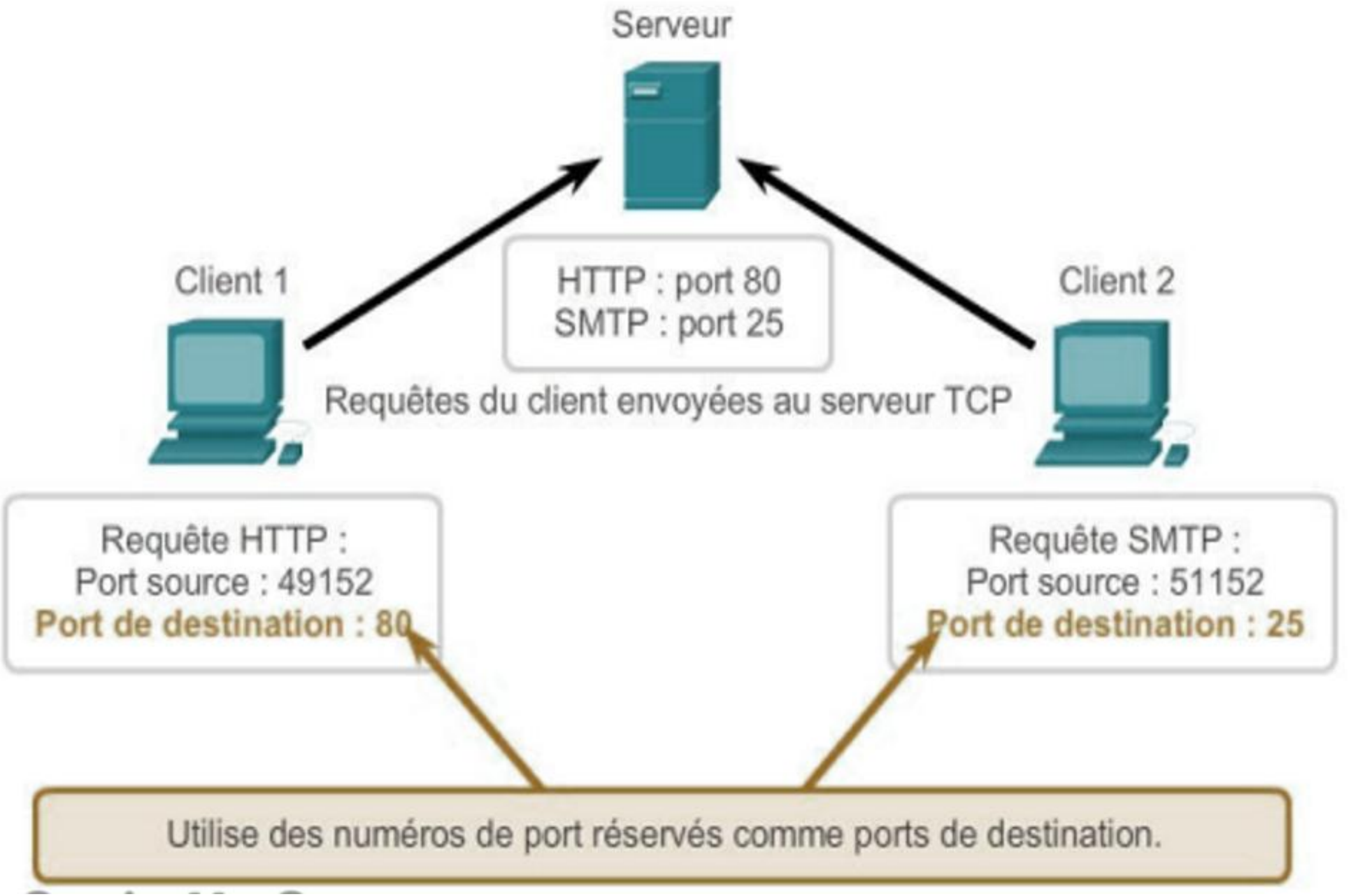
TCP : ports standards

No port	Mot-clé	Description
20	FTP-DATA	File Transfer [Default Data]
21	FTP	File Transfer [Control]
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer
37	TIME	Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
80	HTTP	WWW
110	POP3	Post Office Protocol - Version 3
111	SUNRPC	SUN Remote Procedure Call

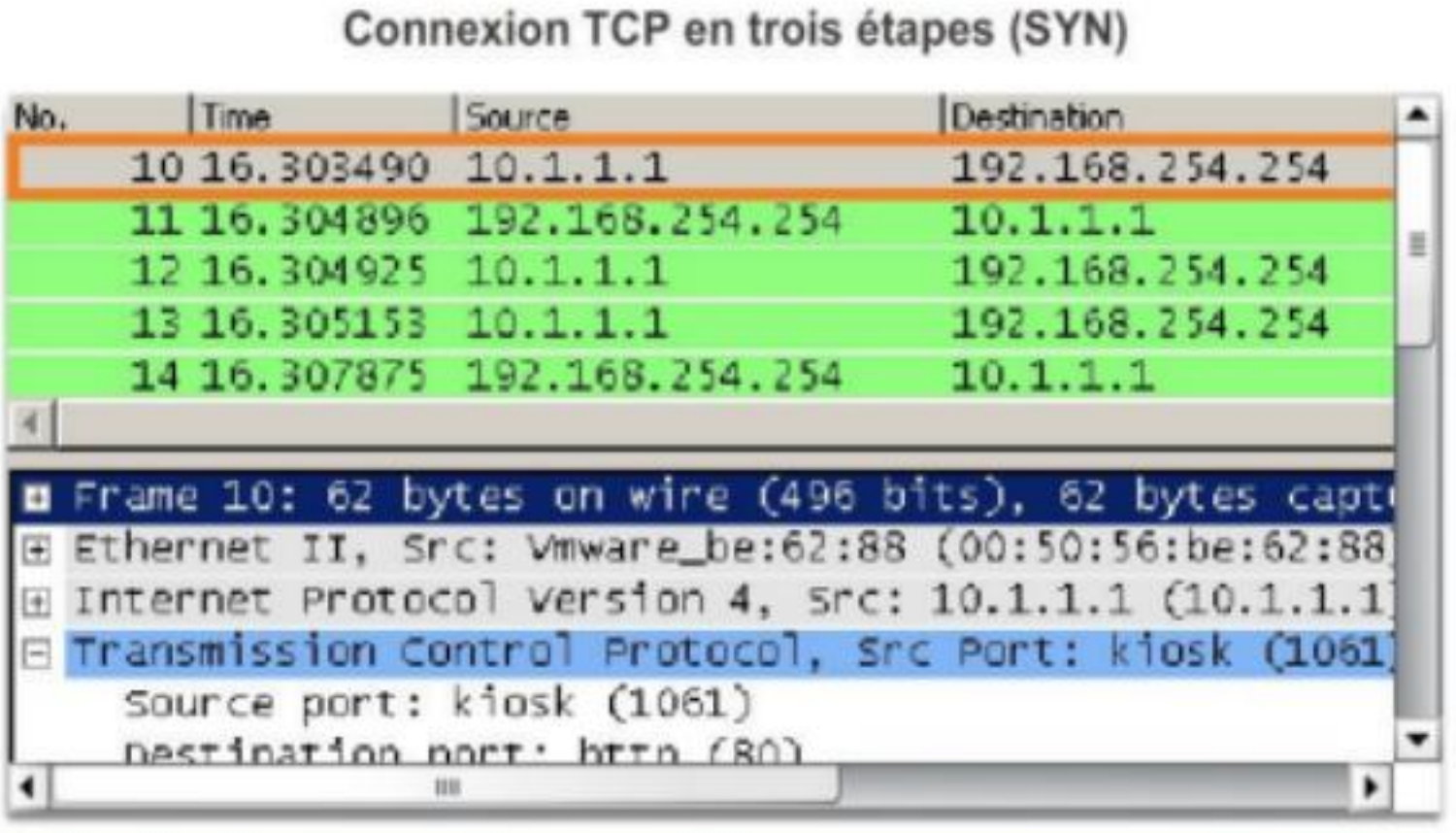
Ports Source des requêtes



Ports de destination des requêtes



Étape 1



L'analyseur de protocole affiche la requête initiale du client pour la session dans la trame 10.

Le segment TCP de cette trame contient les informations suivantes :

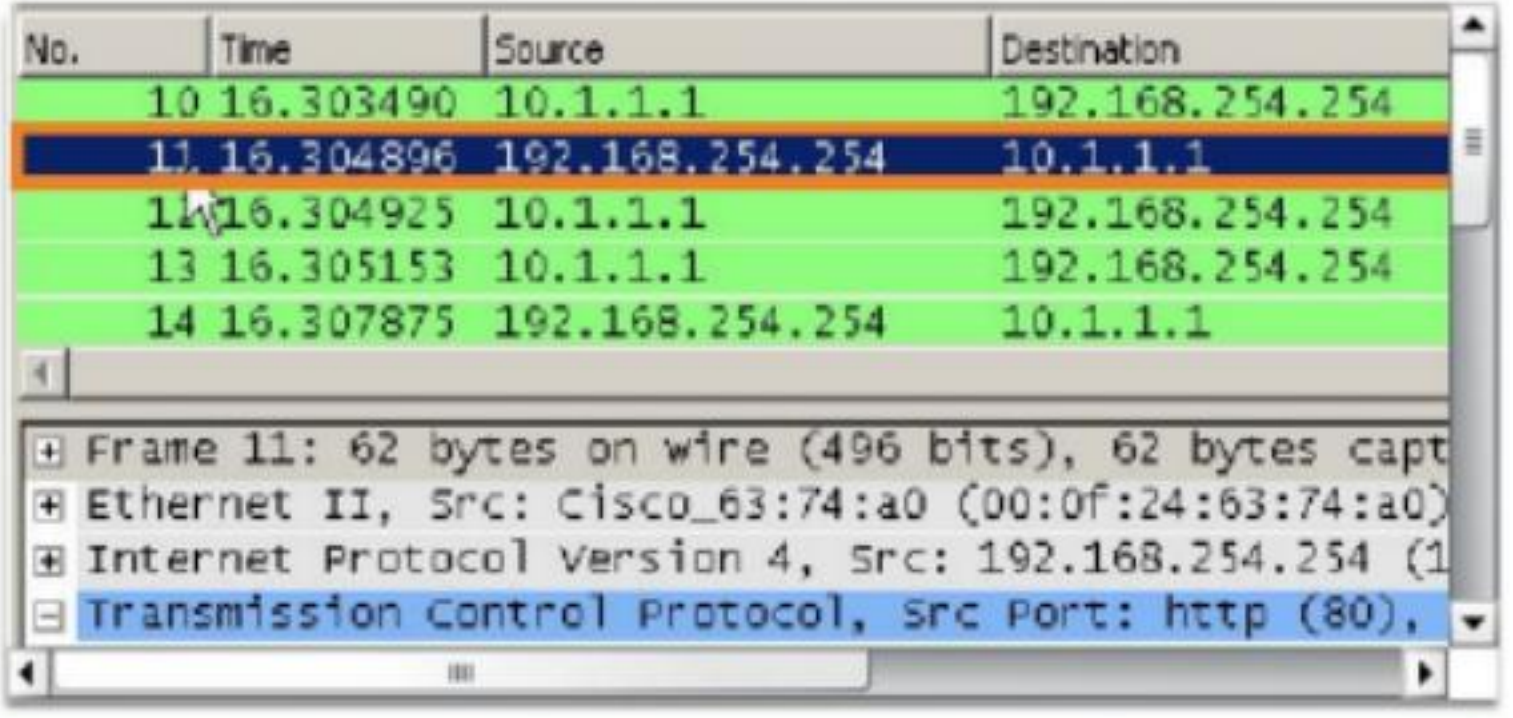
- Indicateur SYN défini pour valider le numéro d'ordre initial (ISN)
- Numéro d'ordre valide sélectionné aléatoirement (la valeur relative est 0)
- Port source sélectionné aléatoirement 1061
- Port de destination réservé 80 (port HTTP), indiquant le serveur Web (httpd)

Le client :

- **demande l'établissement d'une session** de communication client-serveur avec le serveur

Etape 2

Connexion TCP en trois étapes (SYN, ACK)



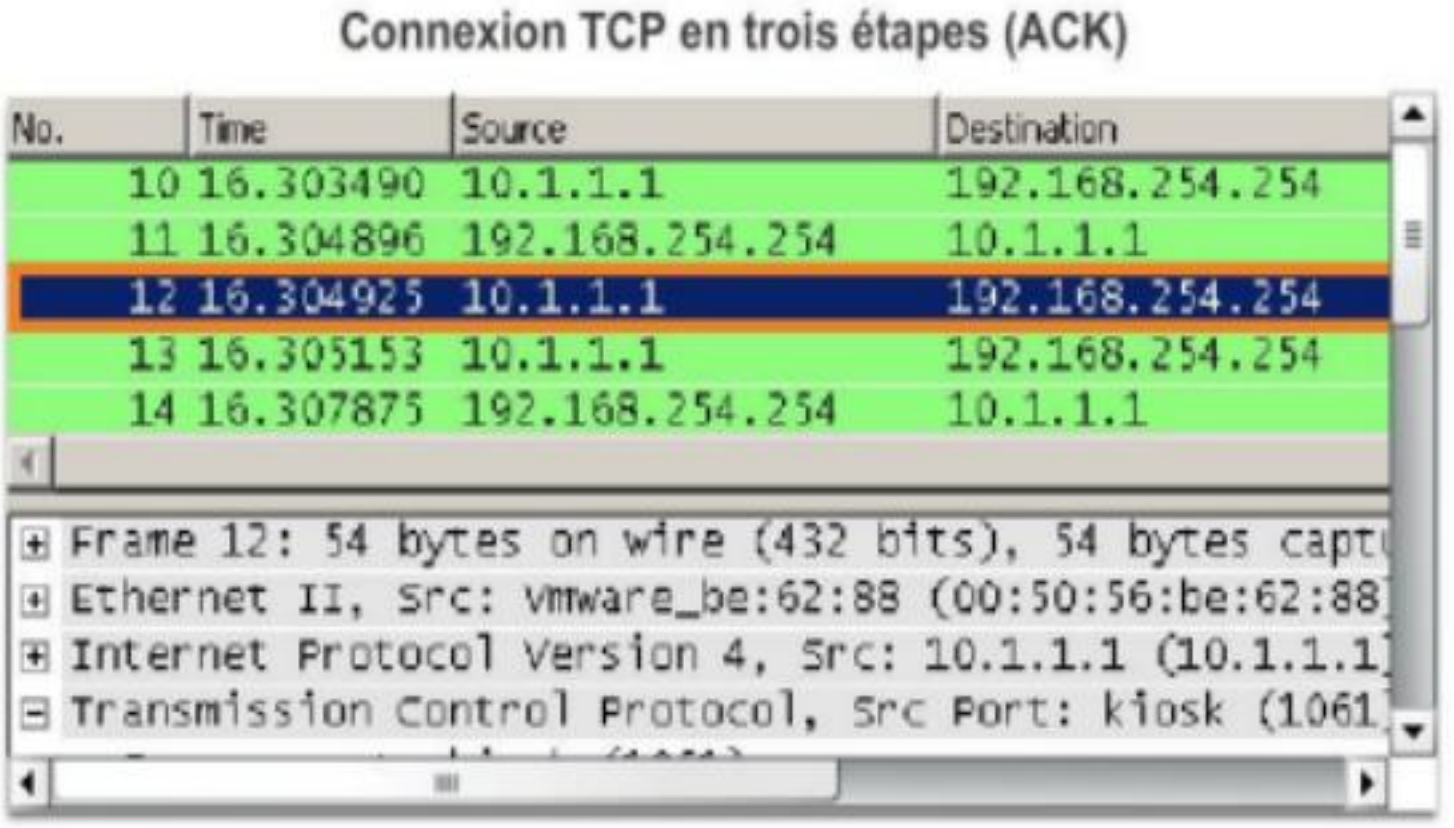
Un analyseur de protocole affiche la réponse du serveur dans la trame 11.

- Indicateur ACK défini pour indiquer un numéro d'accusé de réception valide
- Réponse du numéro d'accusé de réception au numéro d'ordre initial ayant une valeur relative de 1
- Indicateur SYN défini pour indiquer le numéro d'ordre initial (ISN) pour la session du serveur au client
- Numéro de port de destination 1061 correspondant au port source des clients
- Numéro de port source 80 (HTTP) indiquant le service de serveur Web (httpd)

Le serveur :

- **accuse réception de la session** de communication client-serveur
- **demande l'établissement d'une session** de communication serveur-client.

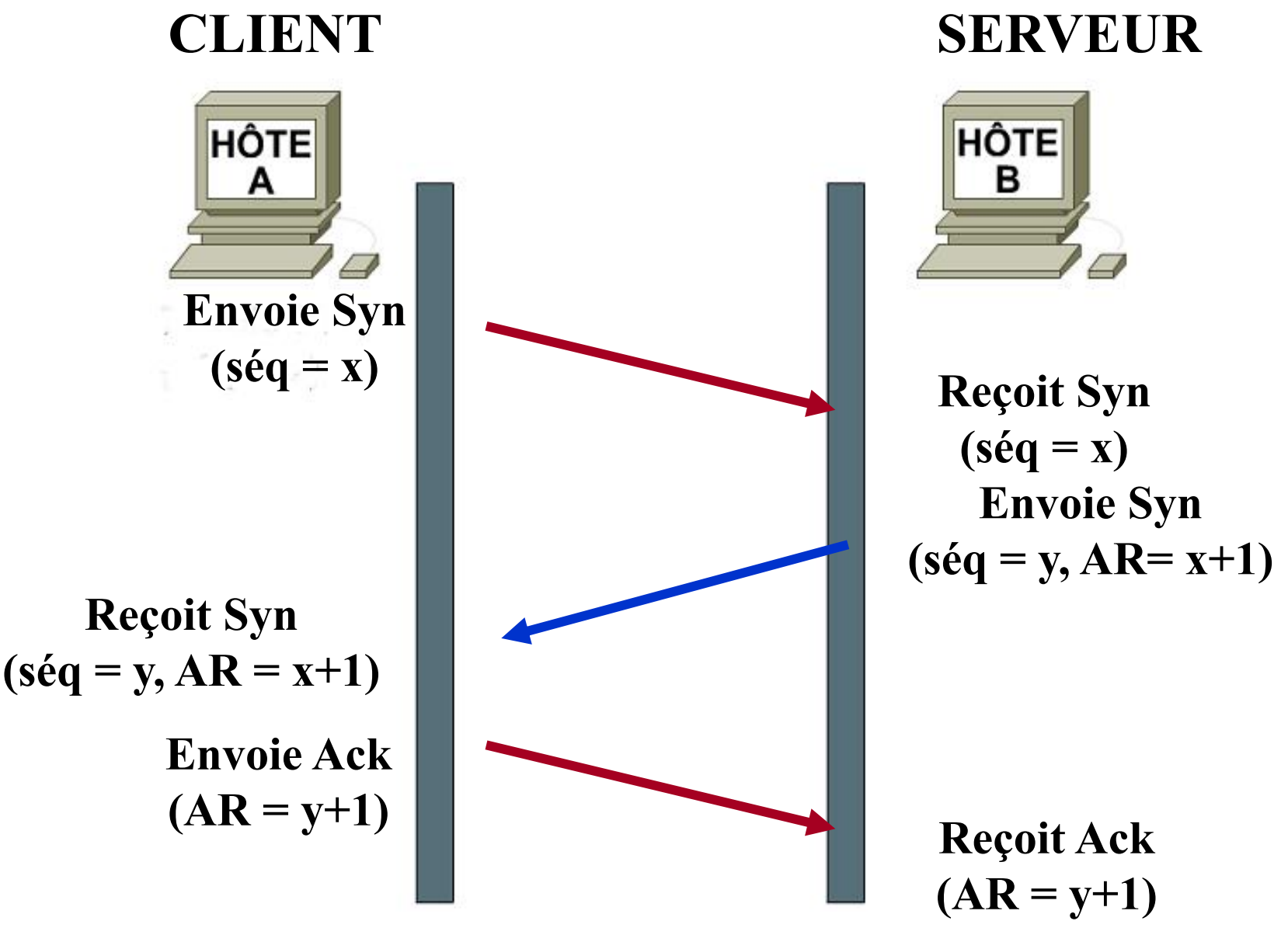
Etape 3



Le client :

- **accuse réception de la session** de communication serveur-client.

Ouverture de connexion



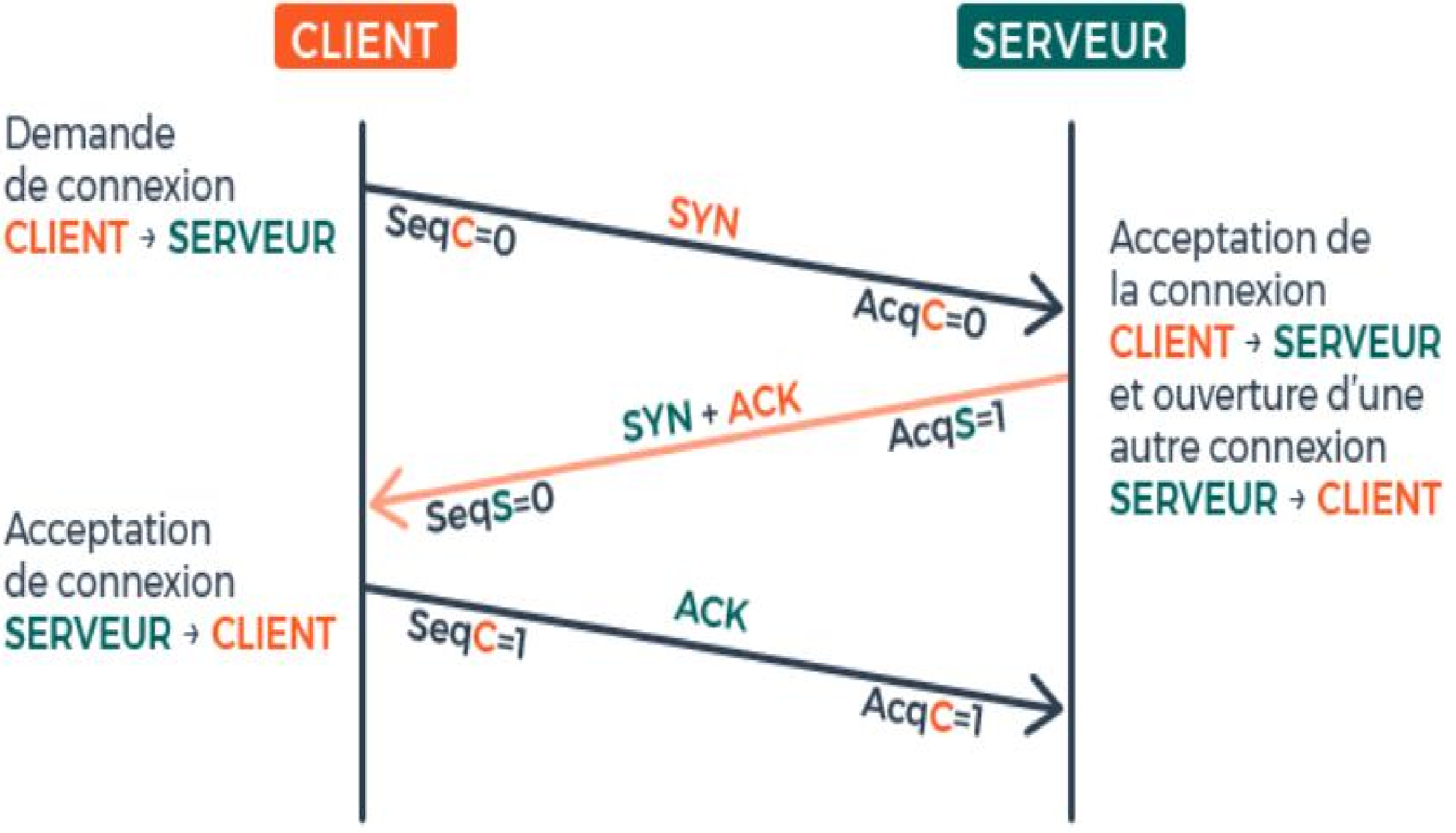
Ouverture de connexion

1. Le client utilise son NumSeq initial dans le champ "**N°séquence**" du segment SYN (**x**).
2. Le serveur ajoute le NumSeq du client plus un (**x+1**) dans le champ "**N°Acquittement**" du segment **ACK**,

3. Le serveur utilise son NumSeq initial dans le champ "**N°séquence**" du segment **SYN** (**y**).

4. Le client confirme en envoyant un segment **ACK** : **N°séquence =** NumSeq du client plus un **(x+1)** et **N°Acquitt** = **N°séquence** du serveur plus un (**y+1**).

verture de connexion



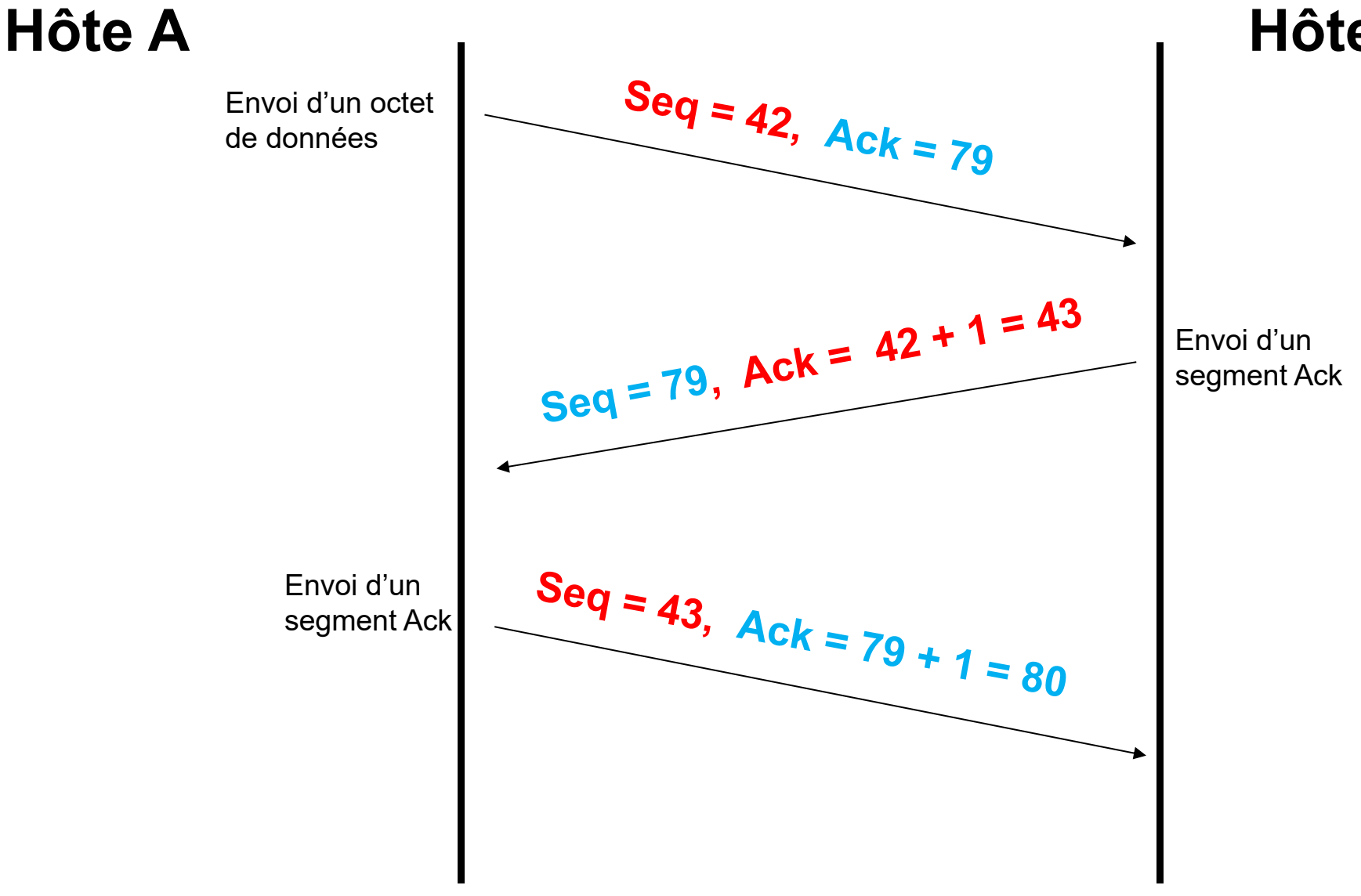
Exemple : échange de segments par Telnet :

1) L'hôte A envoie un **segment SYN** à l'hôte B contenant un octet de données,
    1) un *N° Seq* égal à 42 (**Seq = 42**)
    2) un *N° Ack* égal à 79 (**Ack = 79**),

2) L'hôte B envoie un **segment ACK** à l'hôte A.
    1) Le *N° Seq* de ce segment correspond au *N° Ack* de l'hôte A (**Seq = 79**)
    2) le *N° Ack* au *N° Seq* de A tel que reçu par B, augmenté de la quantité de données en bytes reçue (**Ack = 42 + 1 = 43**),

3) L'hôte A confirme la réception du segment en **envoyant un ACK** à l'hôte B, avec comme
    1) *N° Seq* son nouveau *N° Seq* , à savoir 43 (**Seq = 43**)
    2) *N° Ack* le *N° Seq* du segment précédemment reçu, augmenté de la quantité de données reçue (**Ack = 79 + 1 = 80).**

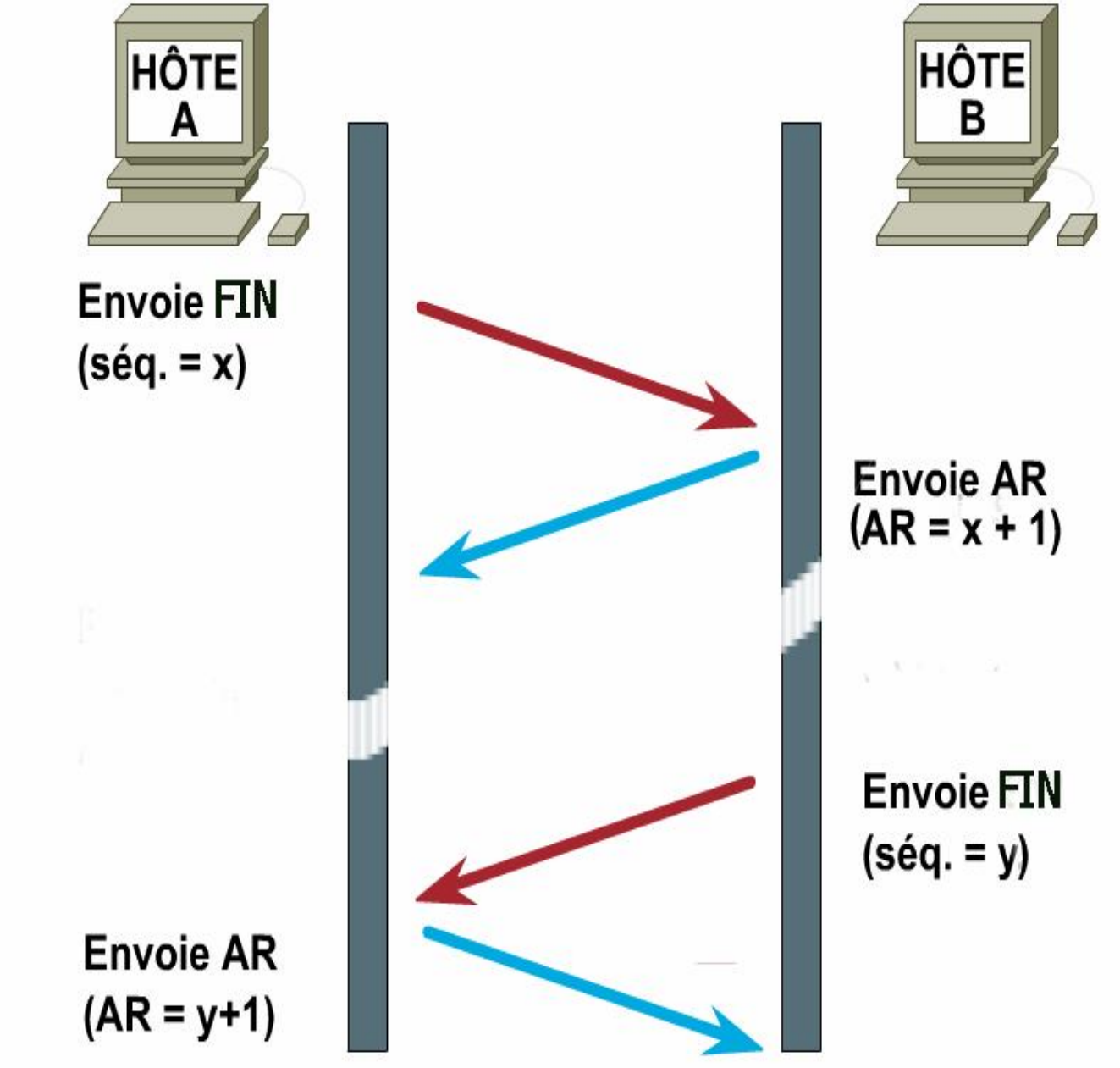
Exemple : échange de segments de données par Telnet :



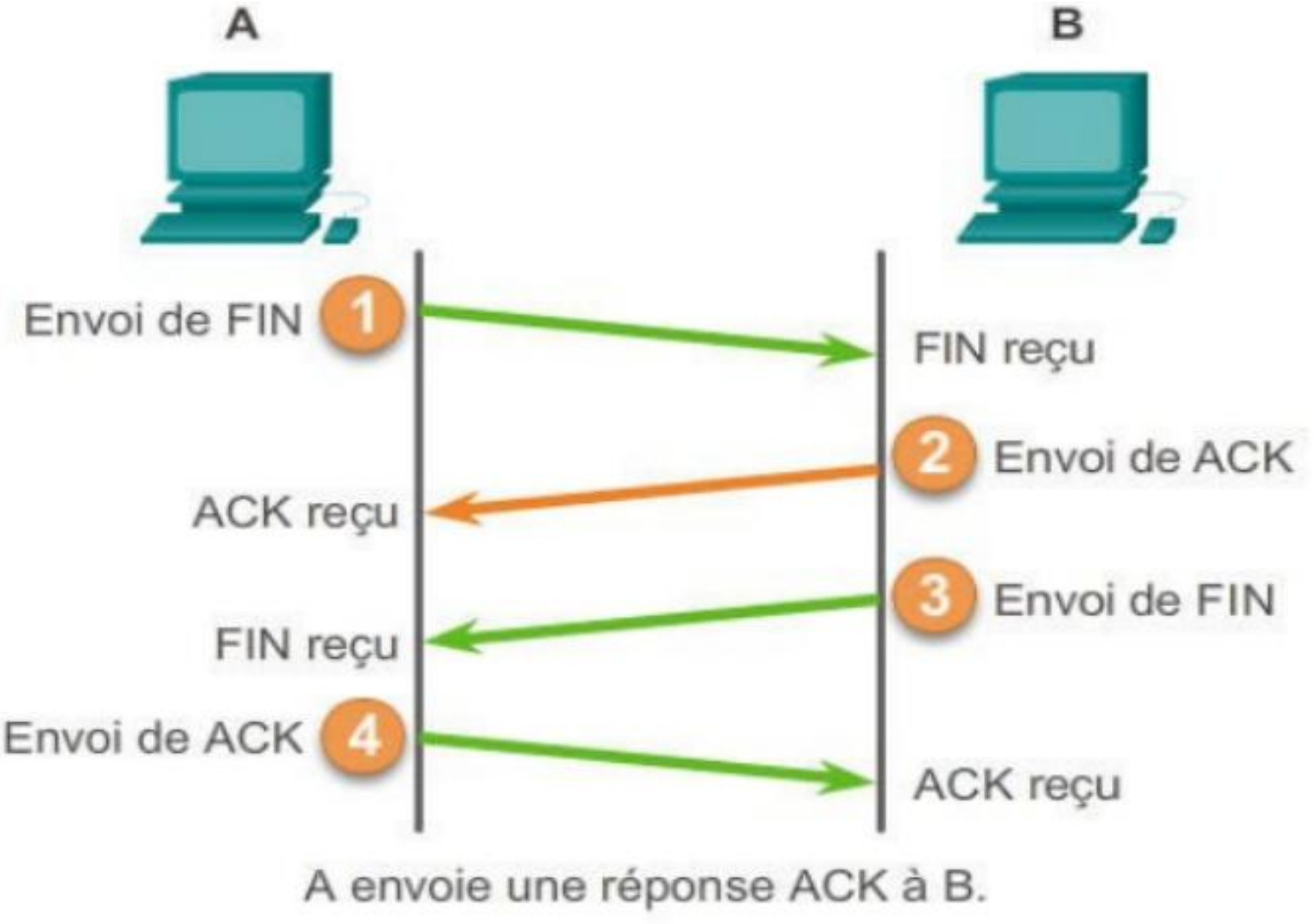
Fin de connexion

chaque extrémité de la connexion effectuant sa terminaison de manière indépendante.

la fin d'une connexion nécessite une paire de segments FIN et ACK pour chaque extrémité.



Fermeture de la session TCP



Fin de connexion

- Sur certains systèmes la clôture se déroule en trois temps :
  - Demande de fin de connexion
  - Acquittement et demande fin de connexion
  - Acquittement

- Possibilité de clore brutalement la connexion par l'envoi d'un segment *RST*

Exemple : échange de segments de données par Telnet :

NB:

- Les NumSeq sont des nombres entiers non signés sur 32 bits, qui reviennent à zéro après avoir atteint 2^32-1.

- Le choix du NumSeq initial est une des clefs de la robustesse et de la sécurité des connexions TCP.

Mécanismes de contrôle du transport

Principes fondamentaux

- Numérotation des segments envoyés.

- Acquittement positif :

  un segment bien reçu doit être acquitté ;

  un segment non acquitté doit être réémis au bout d'un certain temps

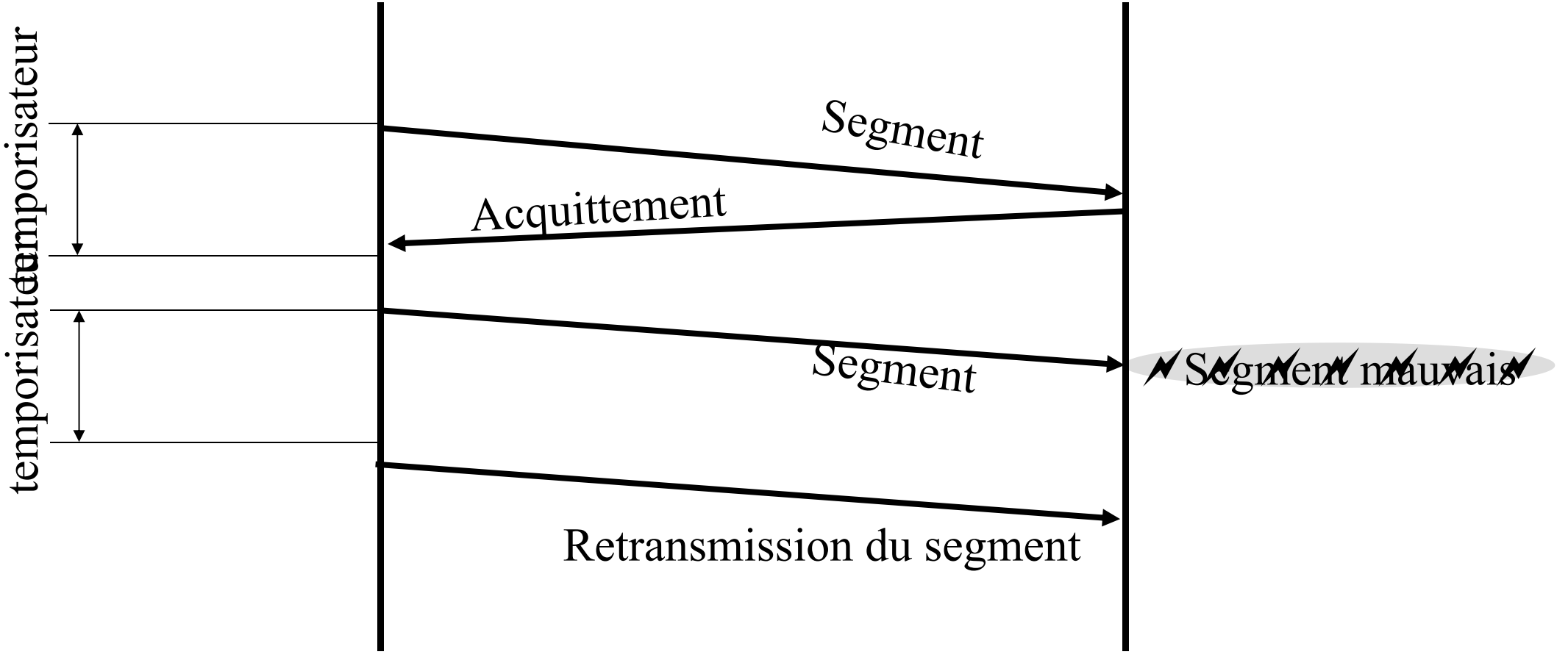
- Acquittement cumulatif et par anticipation

  Utilisation d'une fenêtre glissante dont la taille variera au

  cours de l'échange

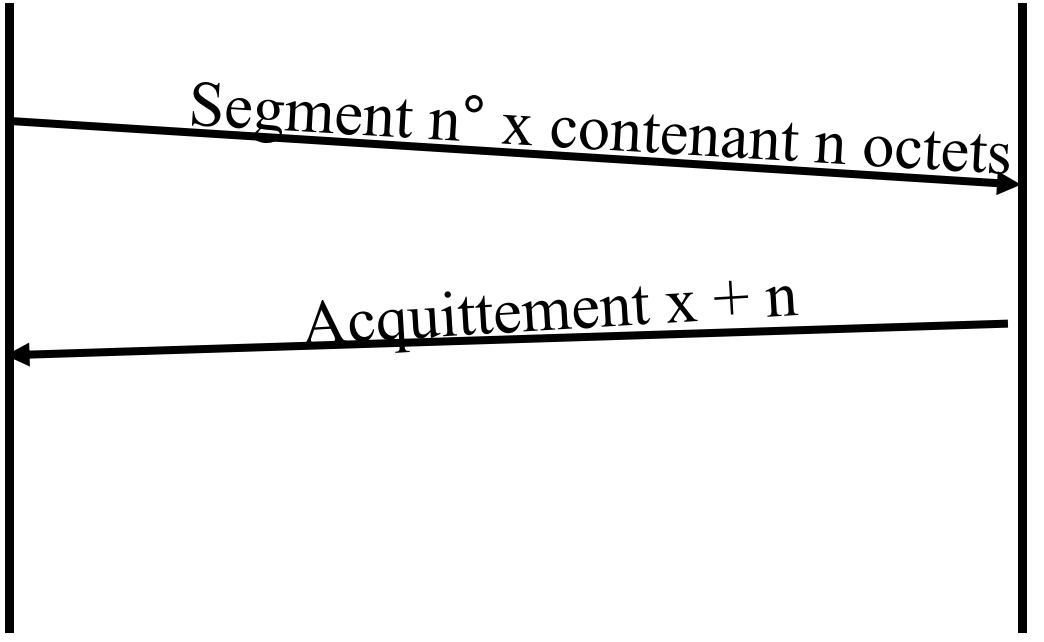
Acquittement positif

- Si le segment n'est pas acquitté, le segment est considéré comme perdu et TCP le retransmet



- TCP gère des **temporisations variables** pour chaque connexion en utilisant un algorithme de retransmission adaptative

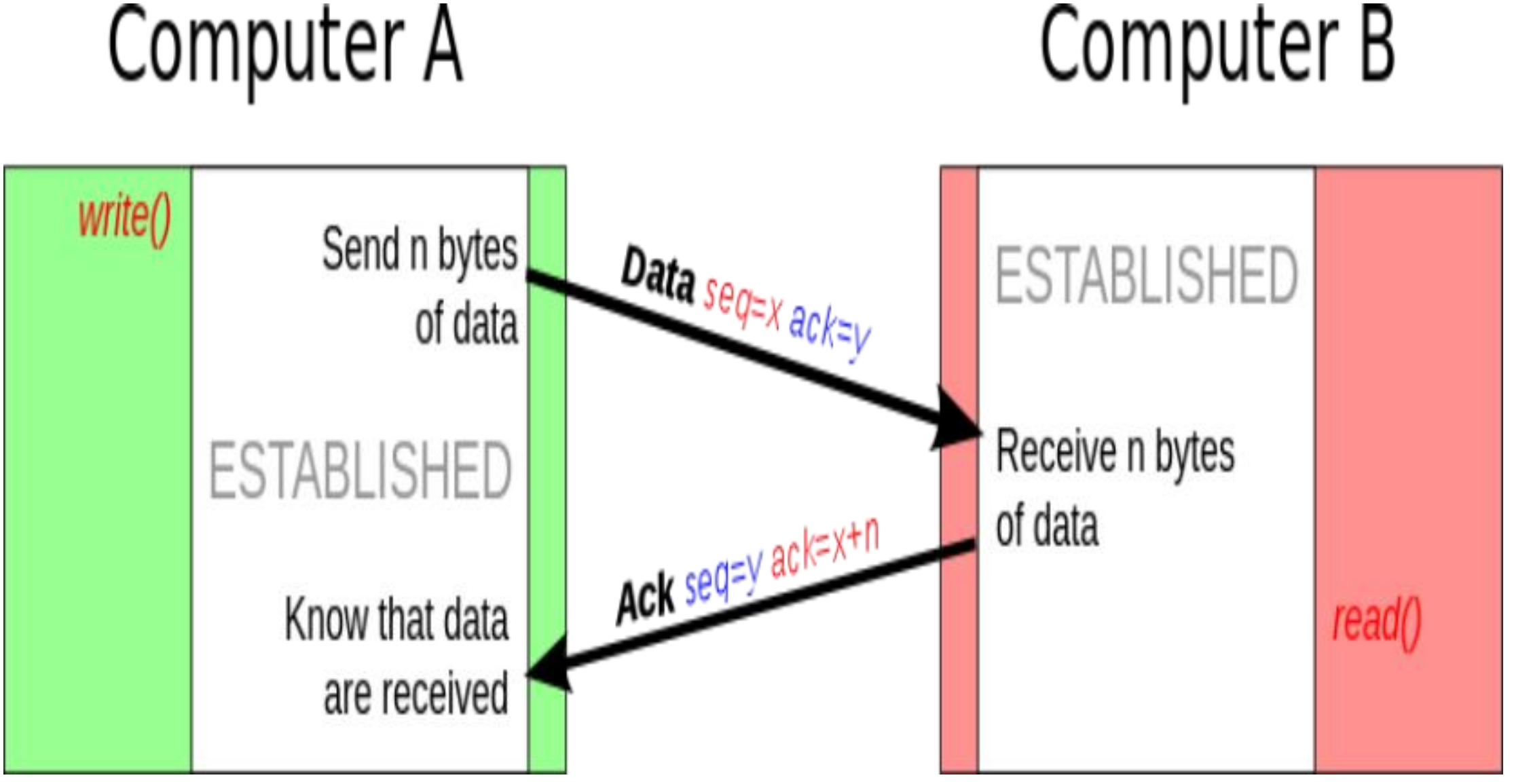
Acquittement cumulatif et par anticipation



- Il indique le N°Seq du prochain octet attendu : tous les octets précédents cumulés sont implicitement acquittés

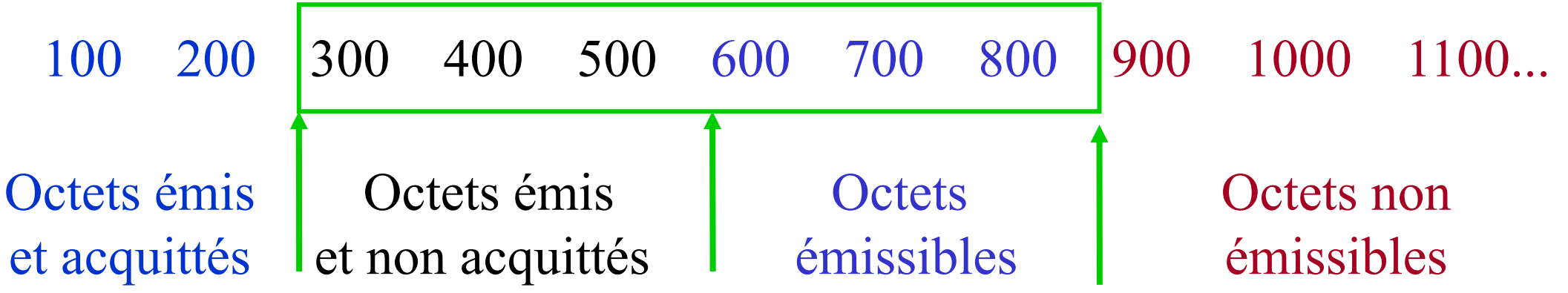
- Si un segment a un N°Seq > N°Seq attendu, le segment est conservé mais l'acquittement référence toujours le N°Seq attendu

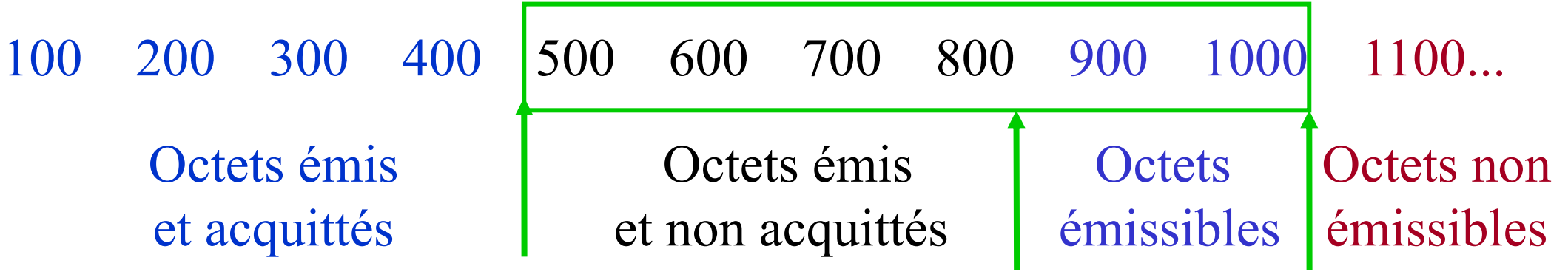
Transfert des données: fenêtre de taille n



Fenêtre glissante

Taille de la fenêtre 6*100 octets
Taille du segment émis 100 octets

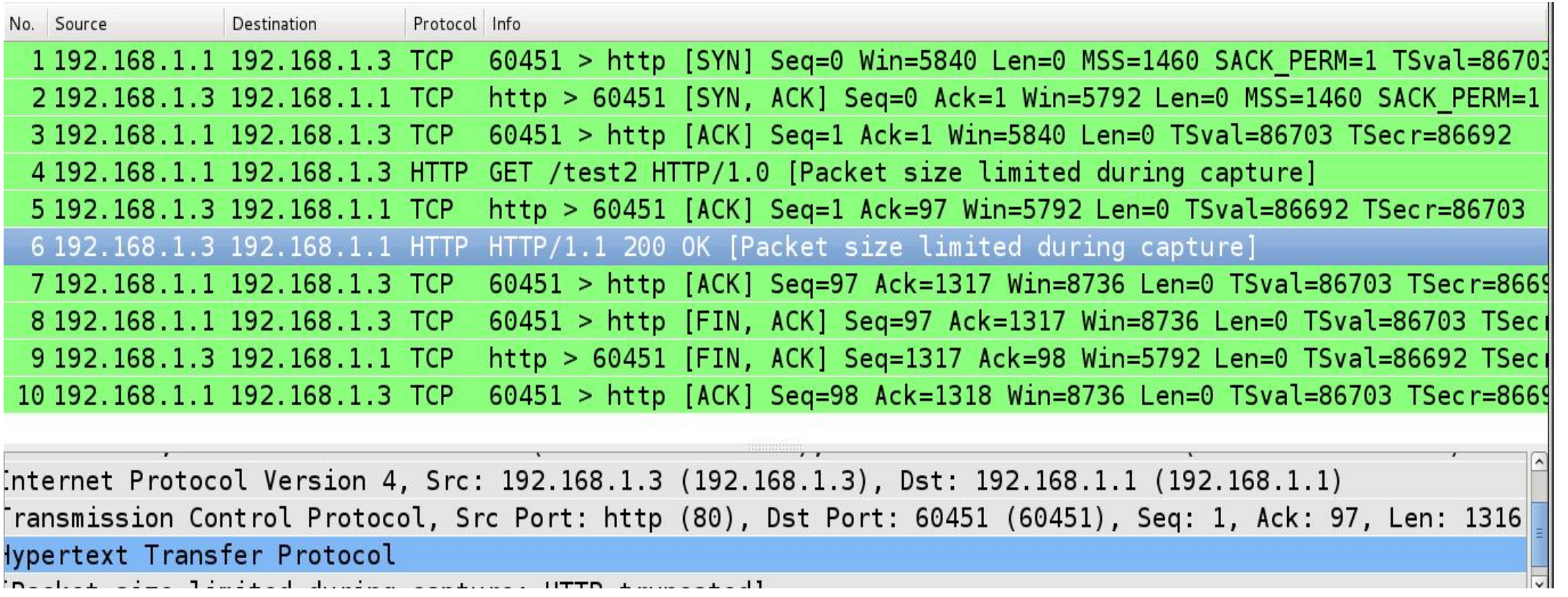




Bilan: Une connexion TCP

1. Ouverture de connexion
    1. Sychronisation
    2. Acknowledge Synchronisation
2. Envoi de trames selon fenêtre disponible
3. Si accusé réception, décaler la fenêtre
4. Si TimeOut, ré-envoyer le segment fautif
5. Envoi trame de fin
6. Accuse réception de la trame de fin

Exemple



No.	Source	Destination	Protocol	Info
1	192.168.1.1	192.168.1.3	TCP	60451 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=86703
2	192.168.1.3	192.168.1.1	TCP	http > 60451 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1
3	192.168.1.1	192.168.1.3	TCP	60451 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=86703 TSecr=86692
4	192.168.1.1	192.168.1.3	HTTP	GET /test2 HTTP/1.0 [Packet size limited during capture]
5	192.168.1.3	192.168.1.1	TCP	http > 60451 [ACK] Seq=1 Ack=97 Win=5792 Len=0 TSval=86692 TSecr=86703
6	192.168.1.3	192.168.1.1	HTTP	HTTP/1.1 200 OK [Packet size limited during capture]
7	192.168.1.1	192.168.1.3	TCP	60451 > http [ACK] Seq=97 Ack=1317 Win=8736 Len=0 TSval=86703 TSecr=866
8	192.168.1.1	192.168.1.3	TCP	60451 > http [FIN, ACK] Seq=97 Ack=1317 Win=8736 Len=0 TSval=86703 TSec
9	192.168.1.3	192.168.1.1	TCP	http > 60451 [FIN, ACK] Seq=1317 Ack=98 Win=5792 Len=0 TSval=86692 TSec
10	192.168.1.1	192.168.1.3	TCP	60451 > http [ACK] Seq=98 Ack=1318 Win=8736 Len=0 TSval=86703 TSecr=866

```
Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.1 (192.168.1.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: 60451 (60451), Seq: 1, Ack: 97, Len: 1316
Hypertext Transfer Protocol
```

- 1-2-3 : connexion
- **4 : requête http (demande d'un fichier)**
- **5 : acquittement**
- **6 : envoie du fichier**
- **7 : acquittement**
- **8-9-10 : déconnexion (en trois temps)**