

Couche Transport

Jean-Luc Damoiseaux
IUT Aix-en-Provence
Département R&T



Plan

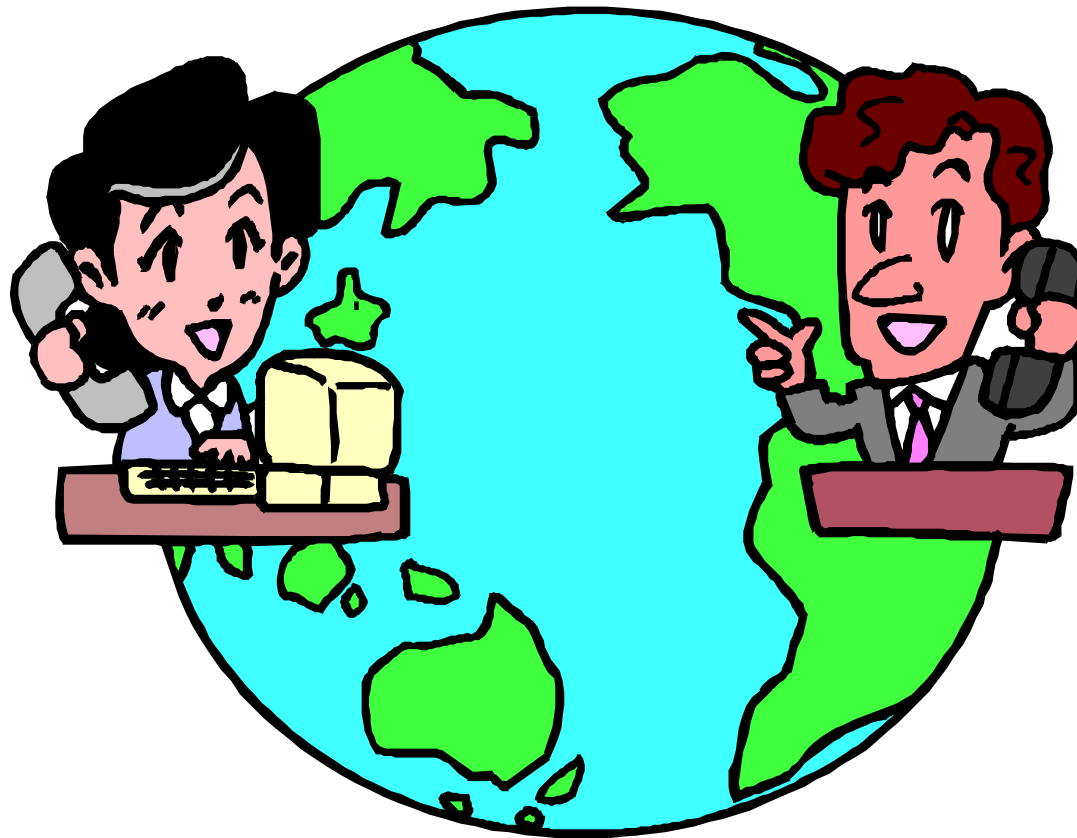
- ⌘ Rappels
 - ⌘ Couche transport
 - ⌘ Types de services
 - ⌘ Le modèle TCP/IP
- ⌘ Etude de TCP
- ⌘ UDP

Couche transport du modèle OSI



- ⌘ Segmente les données de la couche session, et ré-assemble les segments en données pour la couche session
- ⌘ Assure un transport fiable et de bout en bout des segments
 - ⌘ Contrôle le flux d'informations
 - ⌘ Détecte et corrige les erreurs
- ⌘ Est garante de la qualité de service (QOS) rendue par la couche réseau

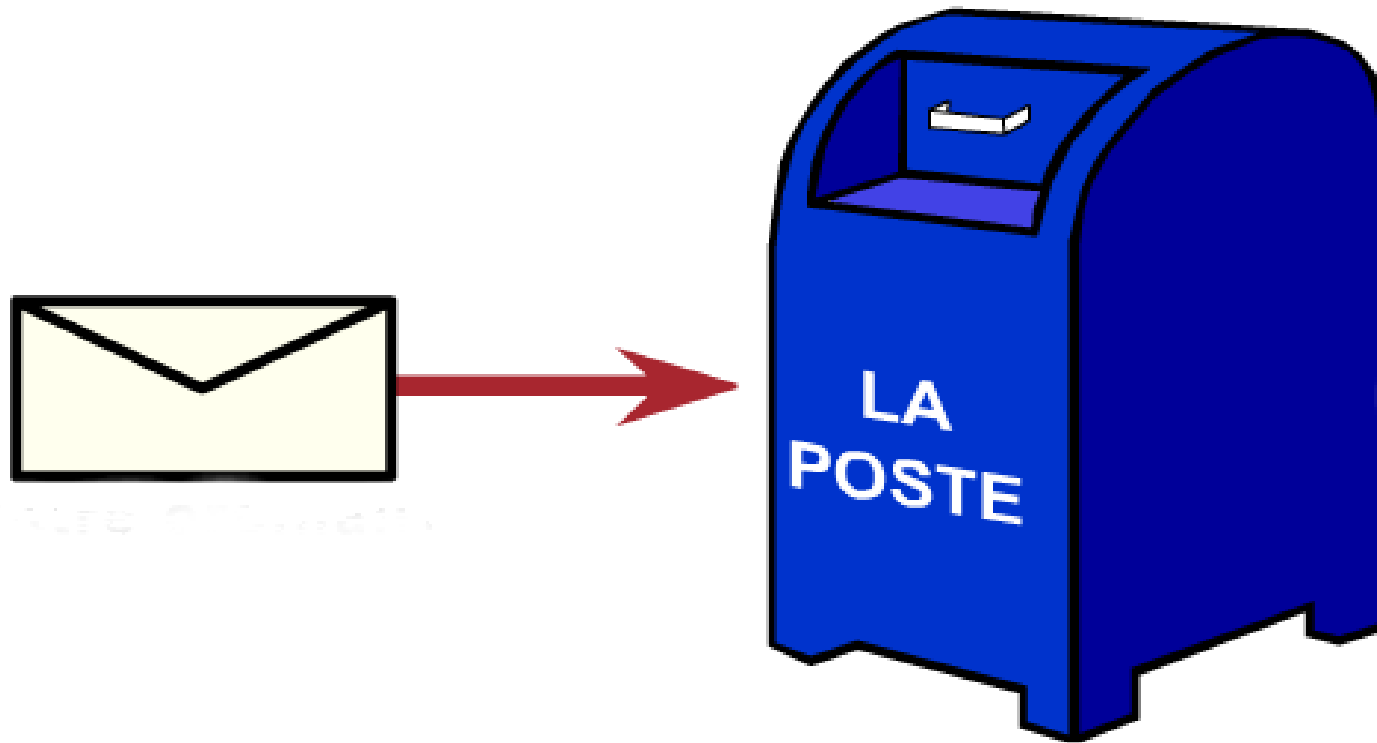
Service mode connecté (1)



Service mode connecté (2)

- ⌘ Fonctionnement en 3 phases
 - ⌘ Etablissement de la connexion entre les entités communicantes
 - ⌘ Transfert des données
 - ⌘ Fermeture de la connexion
- ⌘ Avantages : fiabilité, sécurité et contrôle
- ⌘ Inconvénients : mise en œuvre difficile, gestion du multipoints

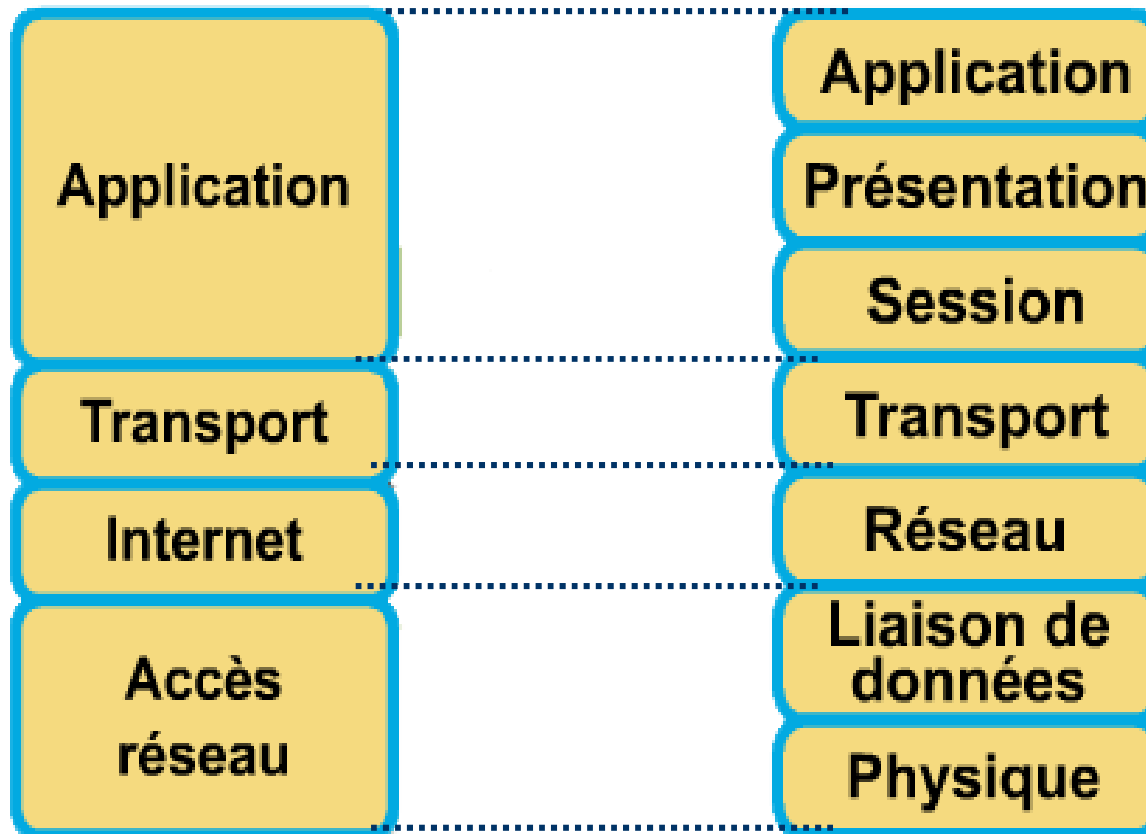
Service mode non connecté (1)



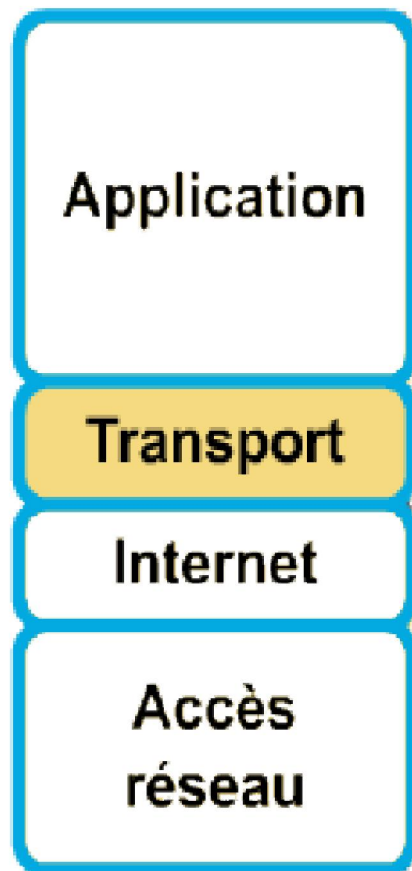
Service mode non connecté (2)

- ⌘ Transfert des données vers le destinataire sans savoir si celui-ci est prêt à les traiter
- ⌘ Avantages : multipoints
- ⌘ Inconvénients : non fiable, pas de contrôle

OSI versus TCP/IP



Couche Transport du modèle TCP/IP



⌘ Protocole avec connexion TCP : gère le contrôle et la qualité du flux, la correction des erreurs

⌘ Protocole sans connexion UDP

Plan

- ⌘ Rappels
- ⌘ Etude de TCP
 - ⌘ Généralités
 - ⌘ Segment TCP
 - ⌘ Notion de port
 - ⌘ Début et clôture de connexion
 - ⌘ Mécanismes de contrôle du transport
- ⌘ UDP

Caractéristiques générales

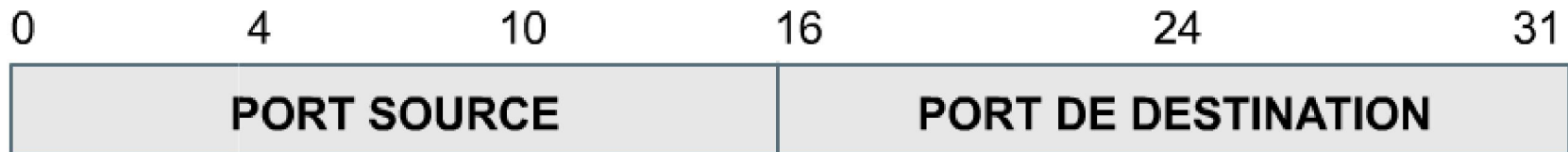
- ⌘ Protocole de bout en bout orienté connexion
- ⌘ Spécifié en 1981 dans le RFC n° 793
- ⌘ Définit un circuit virtuel entre les deux applications qui communiquent
- ⌘ Communication full-duplex

The image features a solid green background on the left side. A white rounded rectangle is cut out from the top-left corner of this green area. A dark blue horizontal bar with rounded ends extends from the right edge of the white shape across the white background.

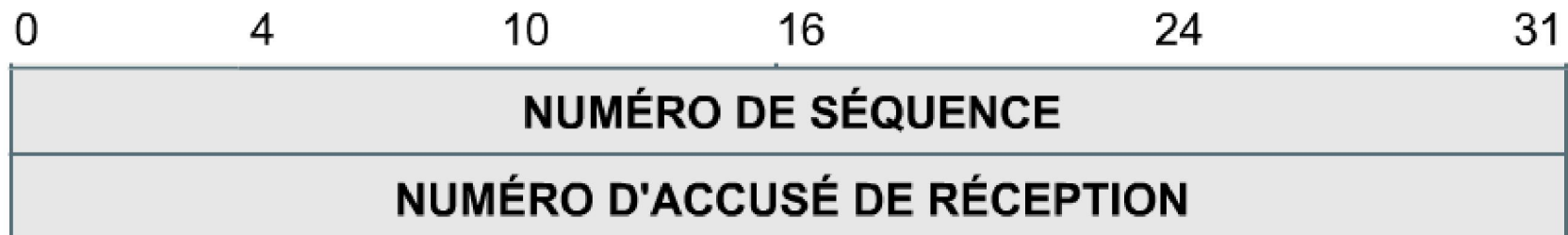
Segment TCP

Structure

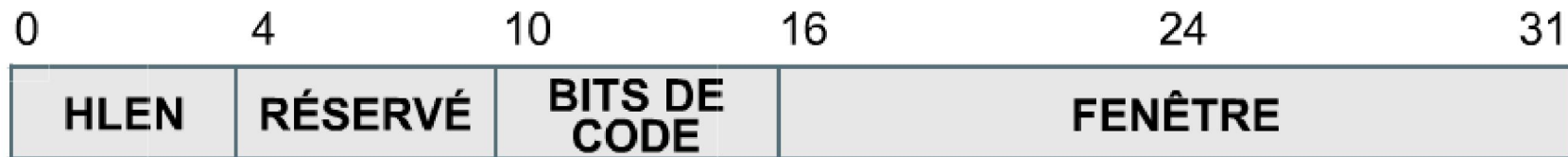
0	4	10	16	24	31
PORT SOURCE			PORT DE DESTINATION		
NUMÉRO DE SÉQUENCE					
NUMÉRO D'ACCUSÉ DE RÉCEPTION					
HLEN	RÉSERVÉ	BITS DE CODE	FENÊTRE		
SOMME DE CONTRÔLE			POINTEUR D'URGENCE		
OPTIONS (S'IL Y A LIEU)				REPLISSAGE	
DONNÉES					
...					



- ⌘ *Port* : entier identifiant l'application à laquelle la couche transport doit remettre les messages



- ⌘ *Numéro de séquence* : identifie la position des données par rapport au segment original
- ⌘ *Numéro d'accusé de réception* : identifie la position du dernier octet reçu dans le flux entrant



- ⌘ *HLEN* : longueur de l'en-tête
- ⌘ *Bits de code* : modificateur du comportement de TCP par caractérisation du segment
- ⌘ *Fenêtre* : nombre d'octets que l'émetteur est prêt à accepter

Bits de code

- ⌘ **URG** : le champ “pointeur d ’urgence” doit être exploité
- ⌘ **ACK** : “numéro d’accusé de réception” doit être exploité
- ⌘ **PSH** : l’émetteur indique au récepteur que toutes les données collectées doivent être transmises à l’application sans attendre les éventuelles données qui suivent
- ⌘ **RST** : Re-initialisation de la connexion
- ⌘ **SYN** : Le champ “numéro de séquence” contient la valeur de début de connexion
- ⌘ **FIN** : L’émetteur du segment a fini d’émettre

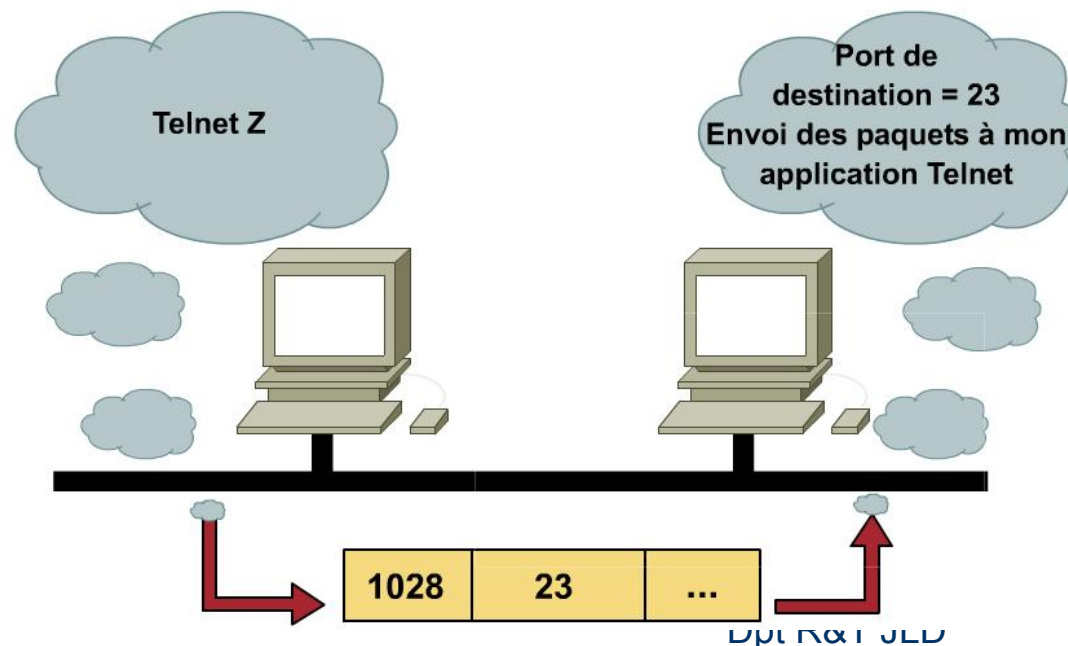


Notion de port

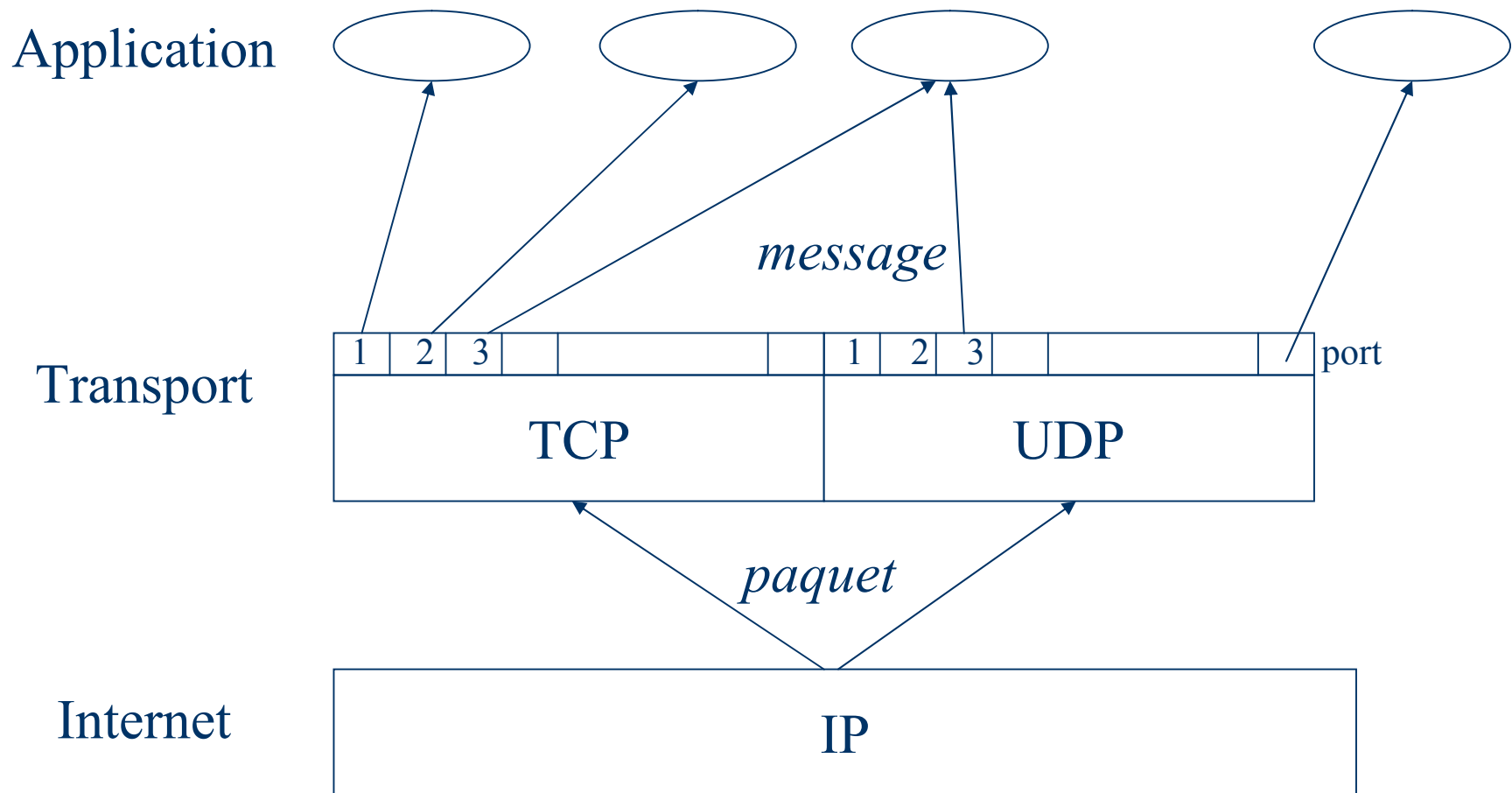


Généralités (1)

- ⌘ Un port est un entier identifiant l'application à laquelle la couche transport doit remettre les messages



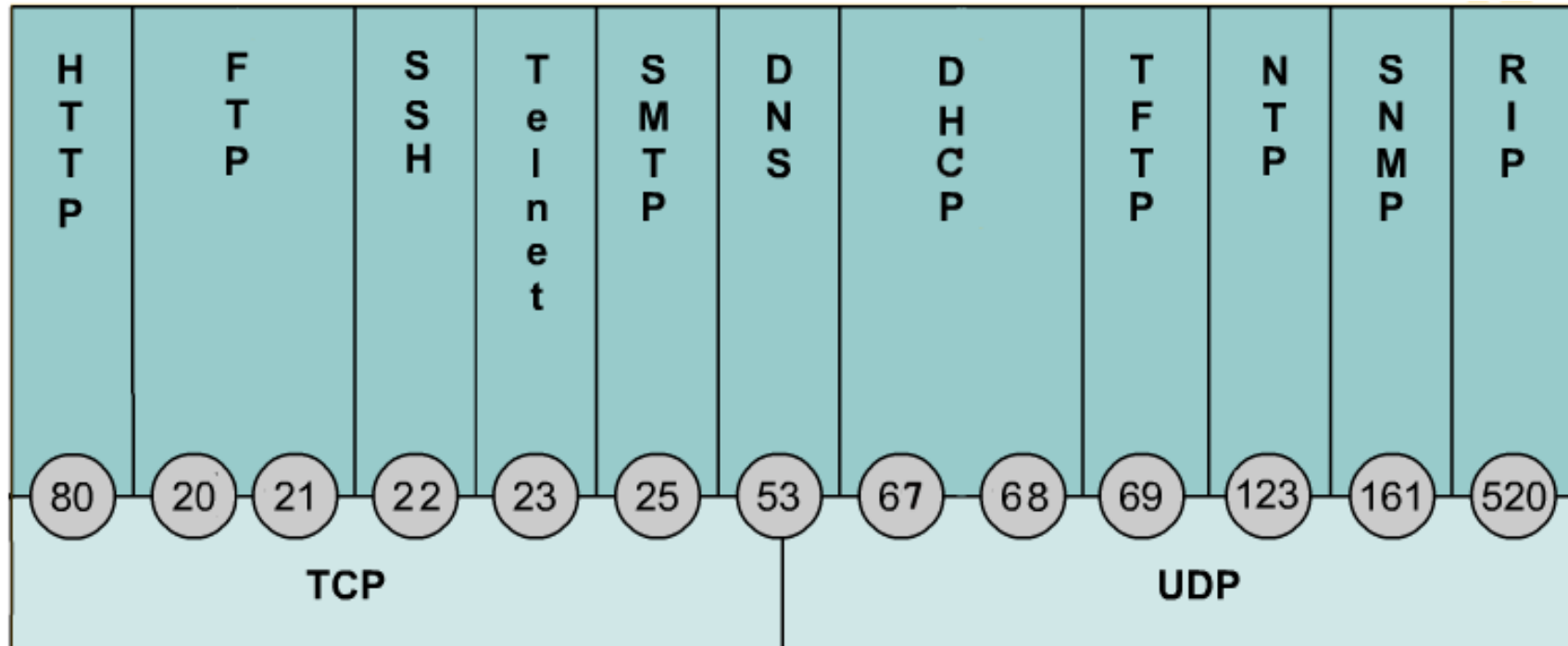
Généralités (2)



Classement des ports

- ⌘ 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

Quelques ports importants

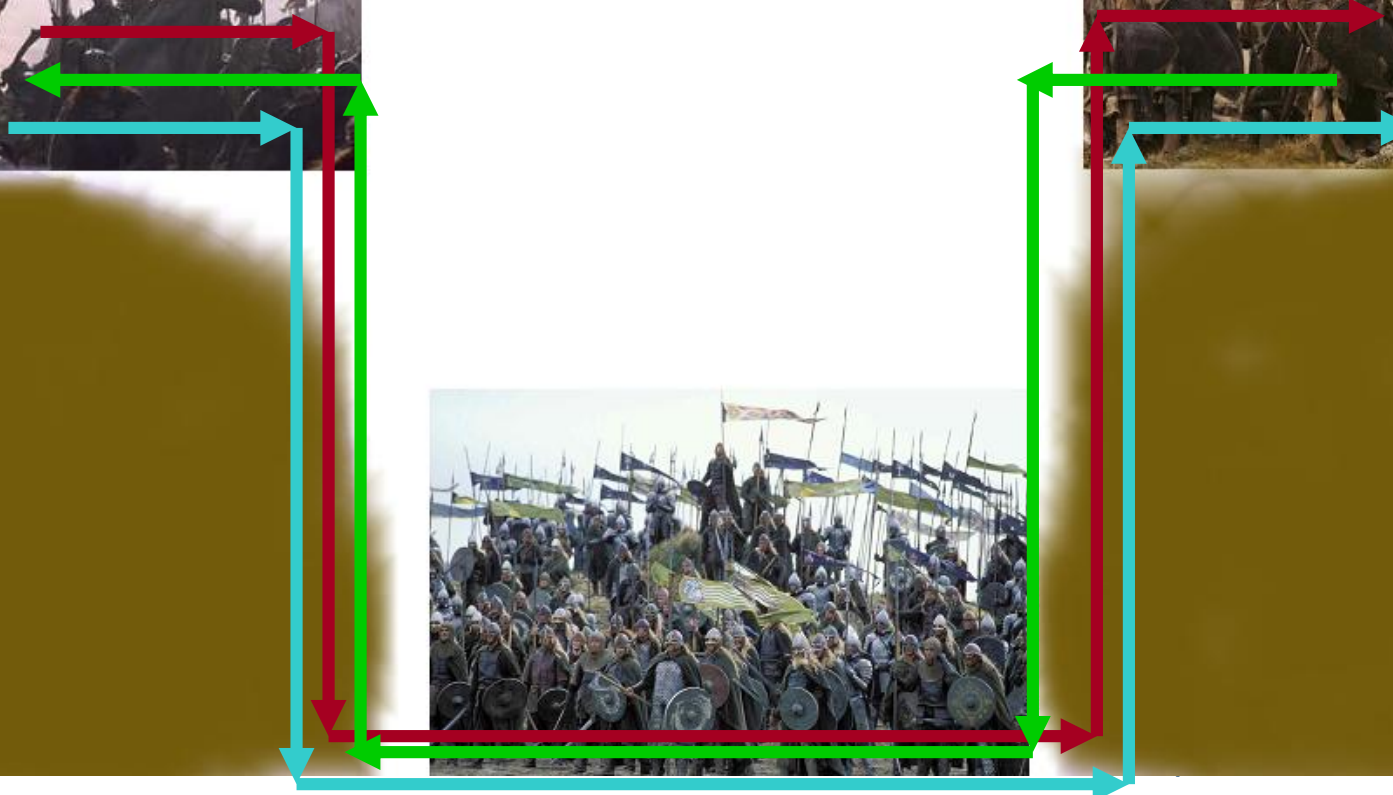
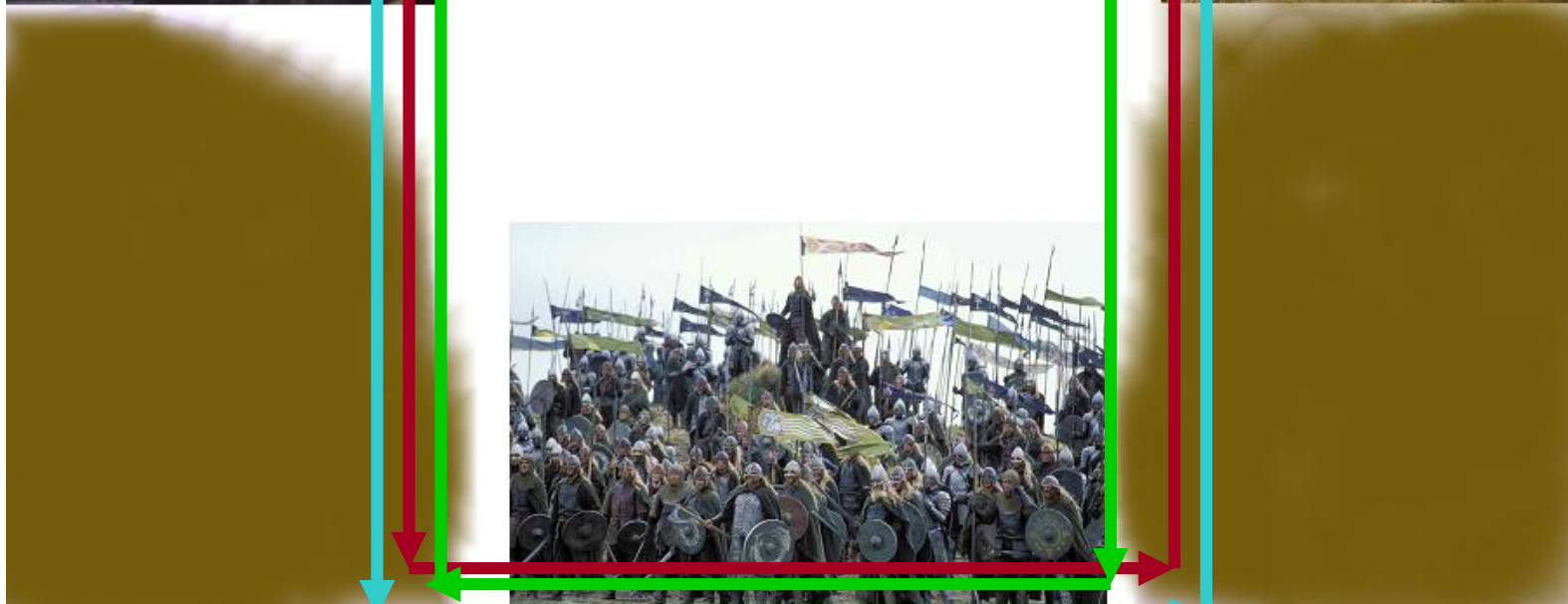
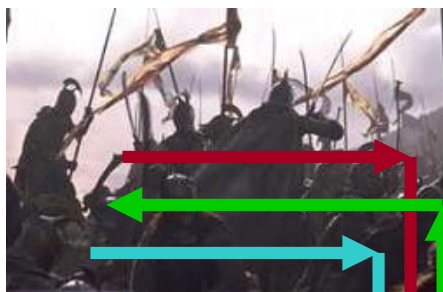




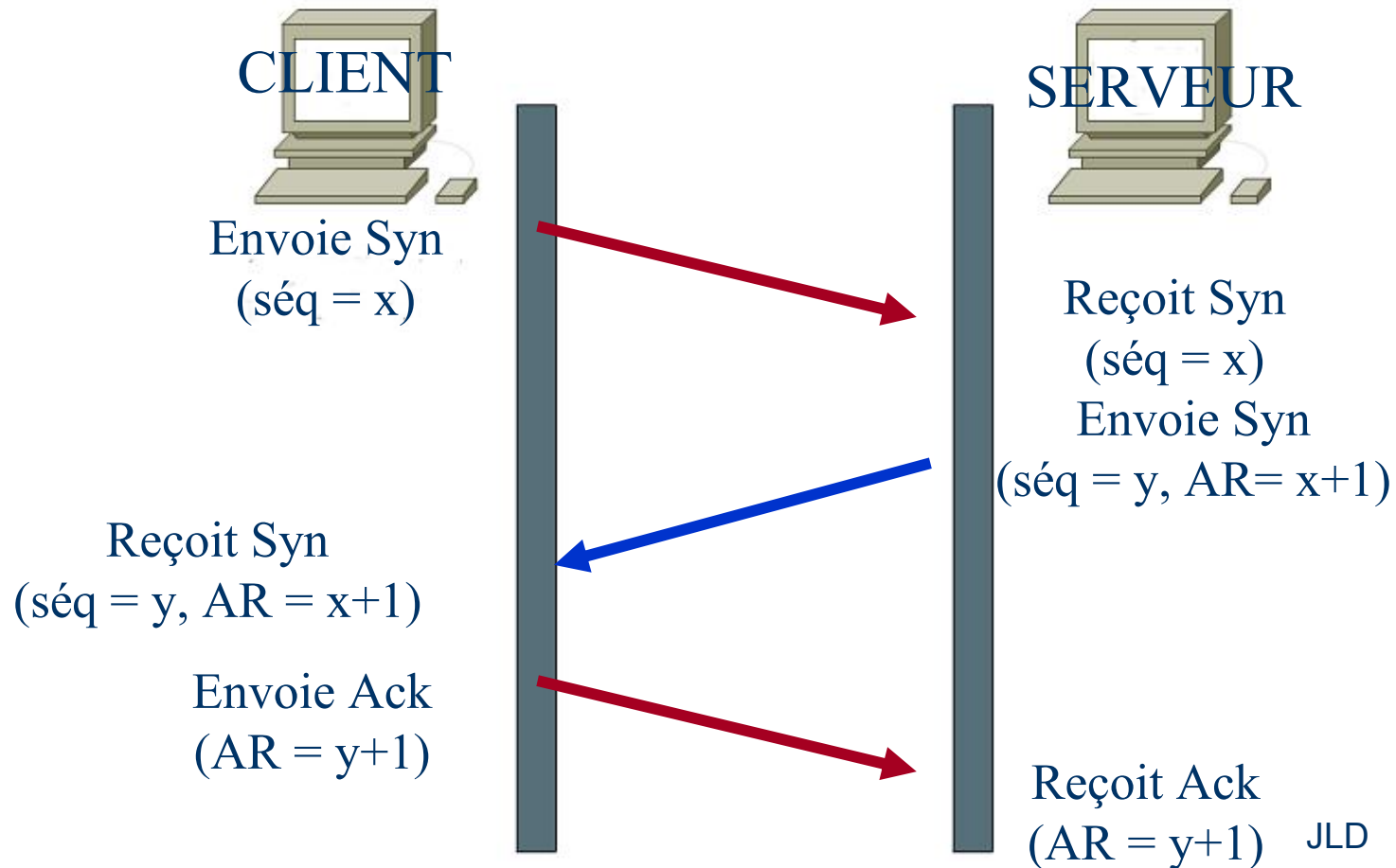
Début et clôture de
connexion



A l'attaque !!!



Ouverture (1)

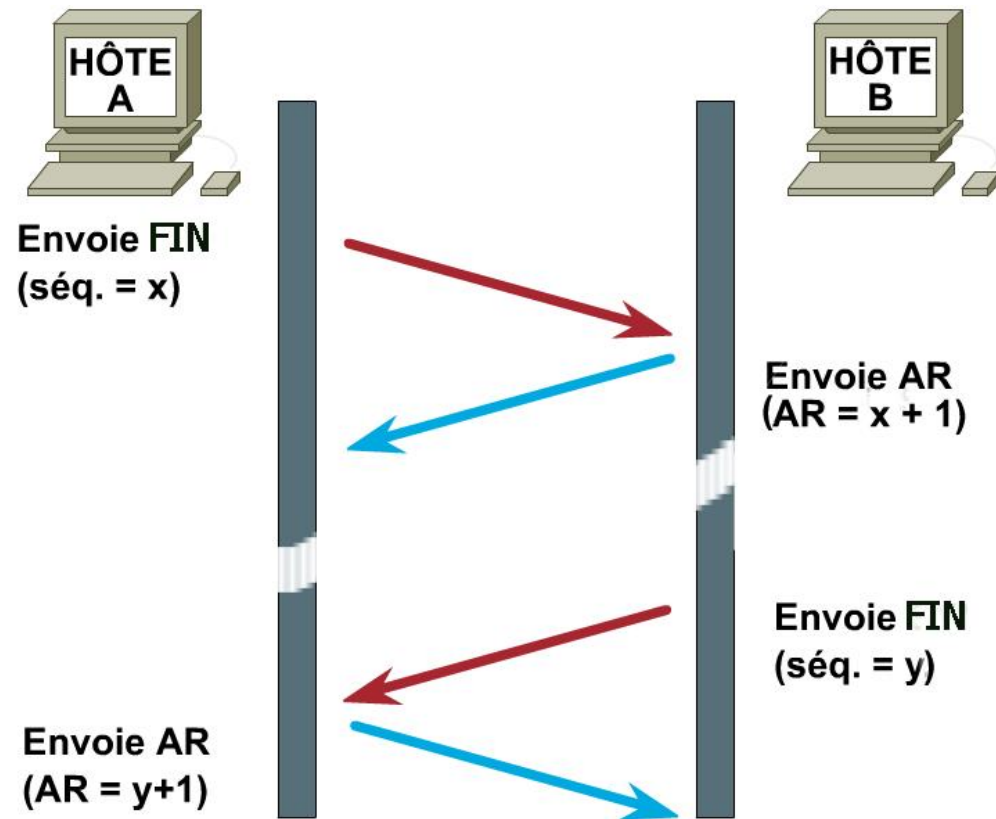


Ouverture (2)

- ⌘ Etablissement de la connexion en trois temps
- ⌘ L'émetteur (appelé client) du premier paquet est à l'origine de l'établissement du circuit ; on parle " d'ouverture active"
- ⌘ Le récepteur (appelé serveur) du premier paquet accepte l'établissement de la connexion ; on parle "d'ouverture passive"

Source	Destination	Protocol	Length	Info
10.1.26.56	10.1.26.187	TCP	74	filenet-rmi > telnet [SYN] Seq=0 Win=5840 Le
10.1.26.187	10.1.26.56	TCP	60	telnet > filenet-rmi [SYN, ACK] Seq=0 Ack=1
10.1.26.56	10.1.26.187	TCP	54	filenet-rmi > telnet [ACK] Seq=1 Ack=1 Win=5
10.1.26.187	10.1.26.56	TELNE	66	Telnet Data ...

Clôture (1)



Clôture (2)

- ⌘ Etablissement de la clôture normale en quatre temps car TCP est Full Duplex
- ⌘ Possibilité de clore brutalement la connexion par l'envoi d'un segment *RST*

Source	Destination	Protocol	Length	Info
10.1.26.187	10.1.26.56	TCP	60	telnet > filenet-rmi [FIN, ACK] Seq=799 Ack=82
10.1.26.56	10.1.26.187	TCP	54	filenet-rmi > telnet [ACK] Seq=82 Ack=800
10.1.26.56	10.1.26.187	TCP	54	filenet-rmi > telnet [FIN] Seq=82 Ack=800
10.1.26.187	10.1.26.56	TCP	60	telnet > filenet-rmi [ACK] Seq=800 Ack=83 Win=4
10.1.26.187	10.1.26.56	TCP	60	telnet > filenet-rmi [FIN, ACK] Seq=799 Ack=82
10.1.26.56	10.1.26.187	TCP	54	filenet-rmi > telnet [FIN, ACK] Seq=82 Ack=800
10.1.26.187	10.1.26.56	TCP	60	telnet > filenet-rmi [ACK] Seq=800 Ack=83 Win=4



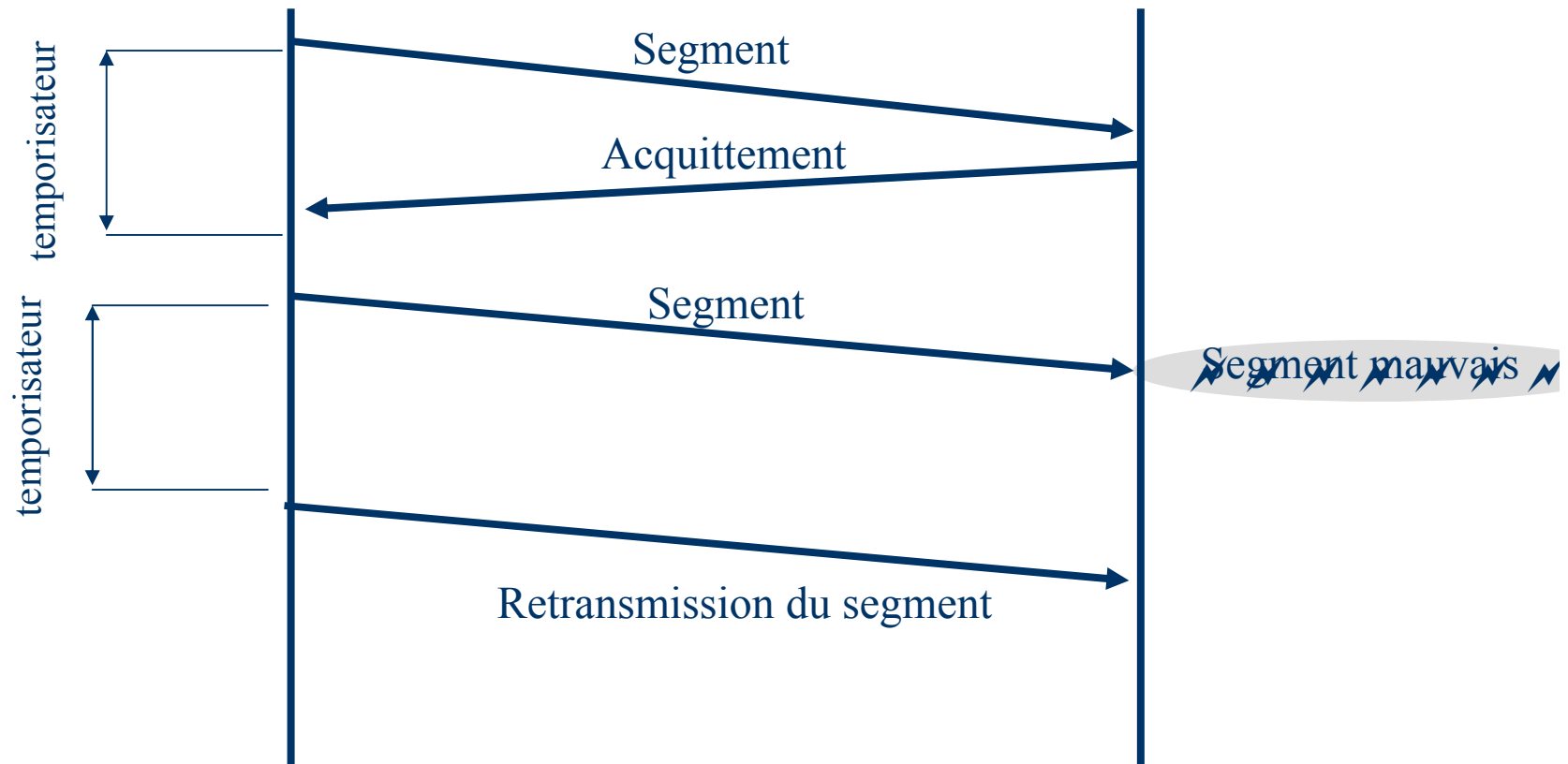
Mécanismes de
contrôle du transport



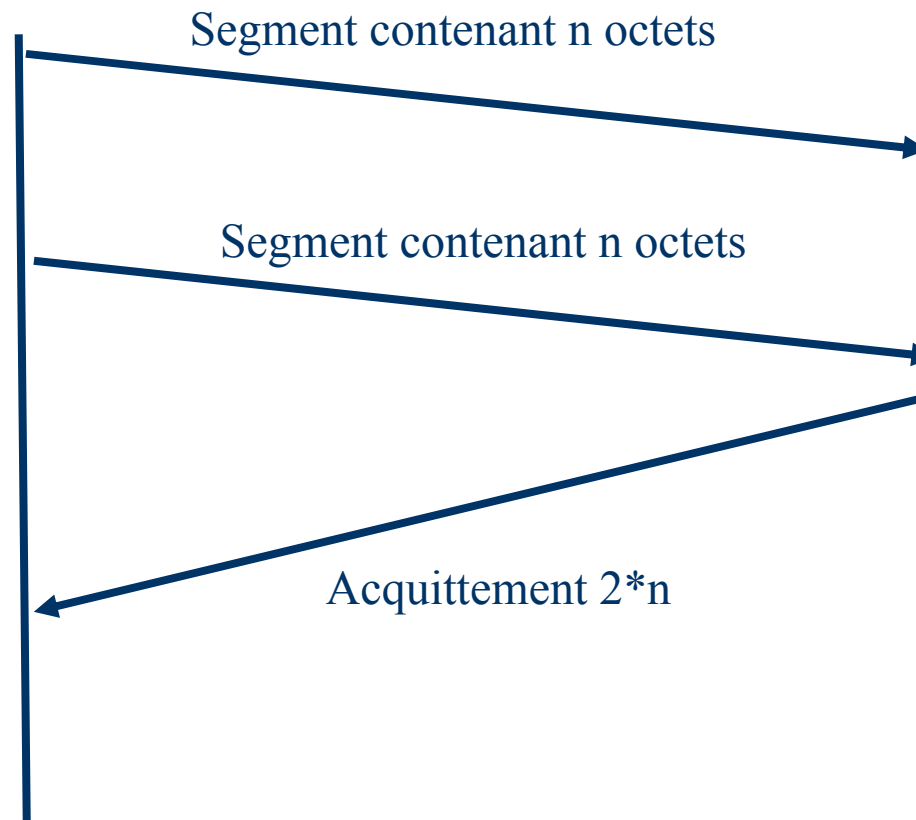
Principes fondamentaux


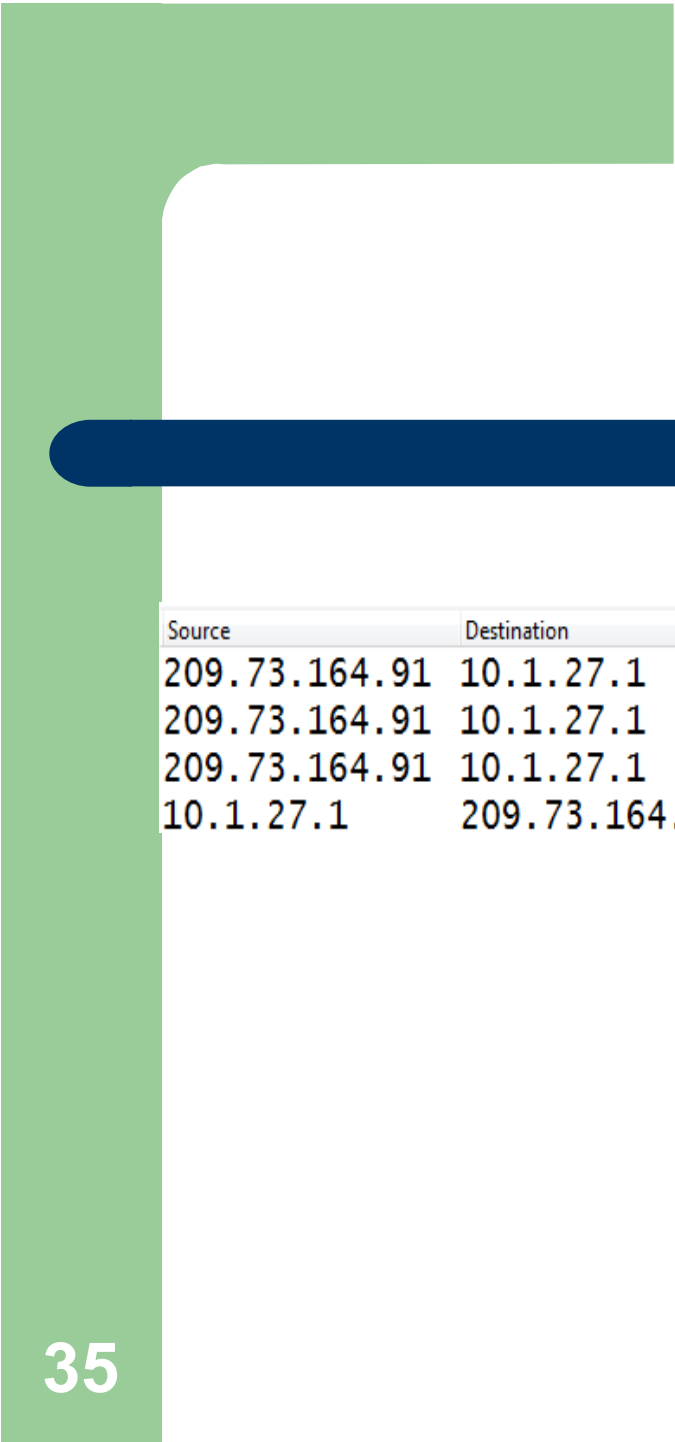
- ⌘ Acquittement positif : un segment bien reçu doit être acquitté ; un segment non acquitté doit être réémis au bout d'un certain temps
- ⌘ Numérotation des segments envoyés
- ⌘ Acquittement cumulatif
- ⌘ Utilisation d'une fenêtre glissante dont la taille variera au cours de l'échange

Acquittement positif



Acquittement cumulatif



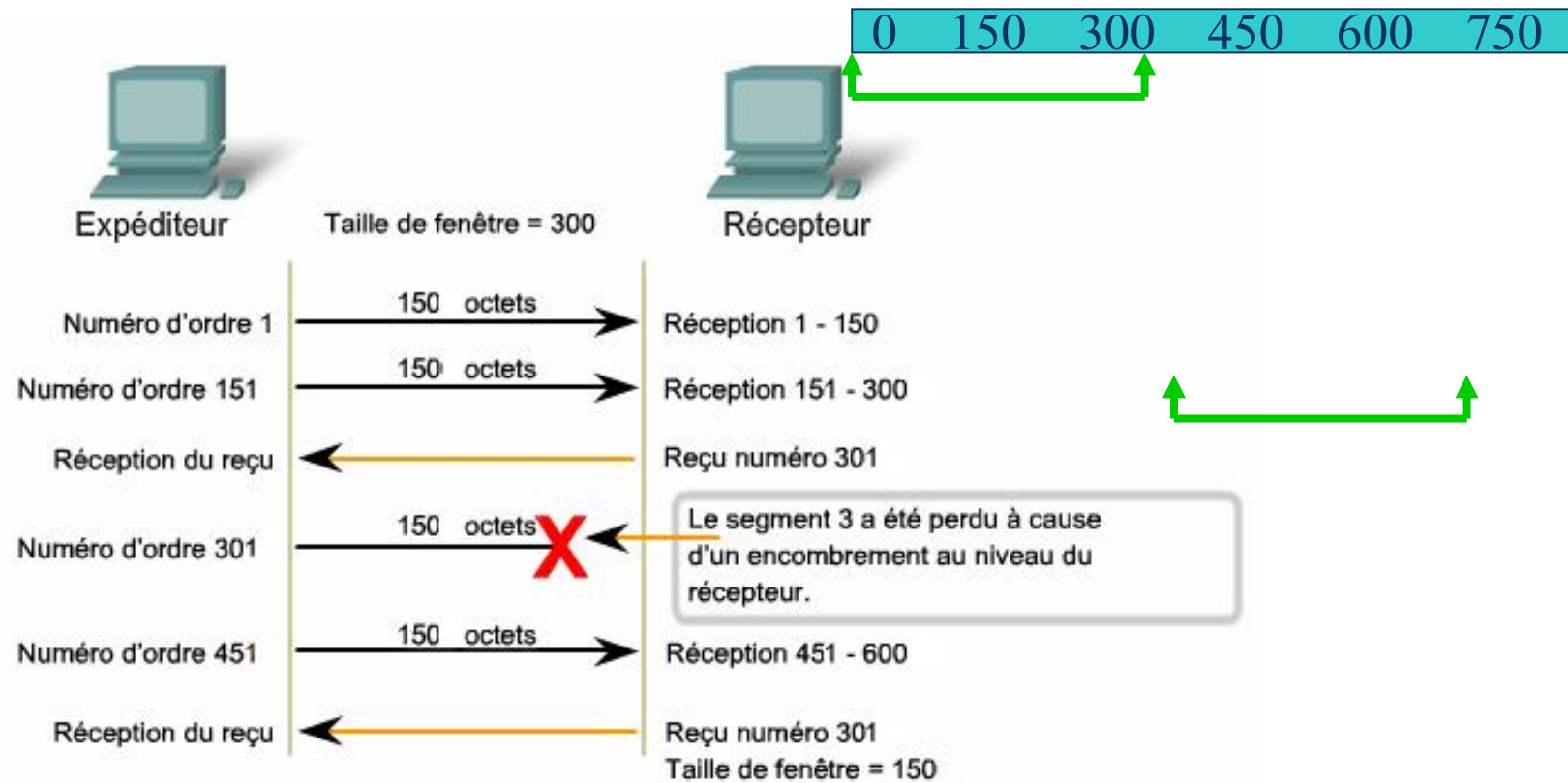


Source	Destination	Protocol	Length	Info
209.73.164.91	10.1.27.1	TCP	60	http > magicnotes [ACK] Seq=1 Ack=359 Len=0
209.73.164.91	10.1.27.1	TCP	1514	[TCP segment of a reassembled PDU] Len=1460
209.73.164.91	10.1.27.1	TCP	1514	[TCP segment of a reassembled PDU] Len=1460
10.1.27.1	209.73.164.91	TCP	54	magicnotes > http [ACK] Seq=359 Ack=2921

Récapitulatif

Source	Destination	Protocol	Info
10.1.26.56	10.1.26.249	TCP	32778 > ftp [SYN] Seq=0 win=5840 Len=0 MSS=1460 TSV=51698
10.1.26.249	10.1.26.56	TCP	ftp > 32778 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460 TSV=51698
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=1 Ack=1 win=5840 Len=0 TSV=51698
10.1.26.249	10.1.26.56	FTP	Response: 220 jumbo.iut.gtr FTP server (Version 6.4/Oper
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=1 Ack=76 win=5840 Len=0 TSV=51698
10.1.26.56	10.1.26.249	FTP	Request: AUTH TLS
10.1.26.249	10.1.26.56	TCP	ftp > 32778 [ACK] Seq=76 Ack=11 win=5792 Len=0 TSV=65796
10.1.26.249	10.1.26.56	FTP	Response: 500 'AUTH TLS': command not understood.
10.1.26.56	10.1.26.249	FTP	Request: USER jld
10.1.26.249	10.1.26.56	FTP	Response: 331- Password
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=21 Ack=132 win=5840 Len=0 TSV=51700
10.1.26.249	10.1.26.56	FTP	Response: 331 Password required for jld.
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=21 Ack=164 win=5840 Len=0 TSV=51700
10.1.26.56	10.1.26.249	FTP	Request: PASS [REDACTED]
10.1.26.249	10.1.26.56	FTP	Response: 230- Linux jumbo 2.4.21 #1 SMP Sat Jun 21 13:5
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=36 Ack=239 win=5840 Len=0 TSV=51700
10.1.26.249	10.1.26.56	FTP	Response: 230-
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=36 Ack=608 win=6432 Len=0 TSV=51700
10.1.26.56	10.1.26.249	FTP	Request: PWD
10.1.26.249	10.1.26.56	FTP	Response: 257 "/export/prof/jld" is current directory.
10.1.26.56	10.1.26.249	FTP	Request: TYPE I
10.1.26.249	10.1.26.56	FTP	Response: 200 Type set to I.
10.1.26.56	10.1.26.249	FTP	Request: PASV
10.1.26.249	10.1.26.56	FTP	Response: 227 Entering Passive Mode (10,1,26,249,146,84)
10.1.26.56	10.1.26.249	FTP	Request: STOR ether_telnet
10.1.26.249	10.1.26.56	FTP	Response: 150 Opening BINARY mode data connection for 'e
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=74 Ack=783 win=6432 Len=0 TSV=51711
10.1.26.249	10.1.26.56	FTP	Response: 226 Transfer complete.
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=74 Ack=807 win=6432 Len=0 TSV=51711
10.1.26.56	10.1.26.249	FTP	Request: QUIT
10.1.26.249	10.1.26.56	TCP	32778 > ftp [FIN, ACK] Seq=80 Ack=807 win=6432 Len=0 TSV=51711
10.1.26.249	10.1.26.56	FTP	Response: 221 Goodbye.
10.1.26.56	10.1.26.249	TCP	32778 > ftp [RST] Seq=81 win=0 Len=0

Fenêtre glissante



Récapitulatif

Source	Destination	Protocol	Info
10.1.26.56	10.1.26.249	TCP	32778 > ftp [SYN] Seq=0 win=5840 Len=0 MSS=1460 TSV=51698
10.1.26.249	10.1.26.56	TCP	ftp > 32778 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0 MSS=1460 TSV=51698
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=1 Ack=1 win=5840 Len=0 TSV=51698
10.1.26.249	10.1.26.56	FTP	Response: 220 jumbo.iut.gtr FTP server (Version 6.4/Oper
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=1 Ack=76 win=5840 Len=0 TSV=51698
10.1.26.56	10.1.26.249	FTP	Request: AUTH TLS
10.1.26.249	10.1.26.56	TCP	ftp > 32778 [ACK] Seq=76 Ack=11 win=5792 Len=0 TSV=65796
10.1.26.249	10.1.26.56	FTP	Response: 500 'AUTH TLS': command not understood.
10.1.26.56	10.1.26.249	FTP	Request: USER jld
10.1.26.249	10.1.26.56	FTP	Response: 331- Password
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=21 Ack=132 win=5840 Len=0 TSV=51700
10.1.26.249	10.1.26.56	FTP	Response: 331 Password required for jld.
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=21 Ack=164 win=5840 Len=0 TSV=51700
10.1.26.56	10.1.26.249	FTP	Request: PASS [REDACTED]
10.1.26.249	10.1.26.56	FTP	Response: 230- Linux jumbo 2.4.21 #1 SMP sat Jun 21 13:5
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=36 Ack=239 win=5840 Len=0 TSV=51700
10.1.26.249	10.1.26.56	FTP	Response: 230-
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=36 Ack=608 win=6432 Len=0 TSV=51700
10.1.26.56	10.1.26.249	FTP	Request: PWD
10.1.26.249	10.1.26.56	FTP	Response: 257 "/export/prof/jld" is current directory.
10.1.26.56	10.1.26.249	FTP	Request: TYPE I
10.1.26.249	10.1.26.56	FTP	Response: 200 Type set to I.
10.1.26.56	10.1.26.249	FTP	Request: PASV
10.1.26.249	10.1.26.56	FTP	Response: 227 Entering Passive Mode (10,1,26,249,146,84)
10.1.26.56	10.1.26.249	FTP	Request: STOR ether_telnet
10.1.26.249	10.1.26.56	FTP	Response: 150 Opening BINARY mode data connection for 'e
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=74 Ack=783 win=6432 Len=0 TSV=51711
10.1.26.249	10.1.26.56	FTP	Response: 226 Transfer complete.
10.1.26.56	10.1.26.249	TCP	32778 > ftp [ACK] Seq=74 Ack=807 win=6432 Len=0 TSV=51711
10.1.26.56	10.1.26.249	FTP	Request: QUIT
10.1.26.249	10.1.26.56	TCP	32778 > ftp [FIN, ACK] Seq=80 Ack=807 win=6432 Len=0 TSV=51711
10.1.26.249	10.1.26.56	FTP	Response: 221 Goodbye.
10.1.26.56	10.1.26.249	TCP	32778 > ftp [RST] Seq=81 win=0 Len=0

ouverture

échange des données

clôture

Plan

- ⌘ Rappels
- ⌘ Etude de TCP
- ⌘ **UDP**

Protocole UDP

- ⌘ User Datagram Protocol est un protocole de transport non orienté connexion et sans garantie de distribution
- ⌘ Spécifié dans le RFC 768
- ⌘ UDP apporte juste un mécanisme de gestion des ports au dessus de la couche réseau ; il est rapide et non fiable

Segment UDP

	Nombre de bits				
16	16	16	16		
Port source	Port de destination	Longueur	Somme de contrôle	Données...	

Divers

- ⌘ Les protocoles utilisant UDP sont les suivants :
 - ⌘ TFTP (trivial file transfert protocol)
 - ⌘ SNMP (simple network managment protocol)
 - ⌘ NFS (*Network File System*)
 - ⌘ Système de noms de domaine (DNS)