

Chapitre 3 :

Protocoles de Transport

Couche Transport

- La couche transport permet de faire communiquer directement deux ou plusieurs entités sans avoir à se préoccuper des différents éléments de réseaux traversés
 - abstraction de la topologie et des technologies sous-jacentes
- La couche transport gère le *fractionnement* et le *réassemblage* en paquets du flux de données à transmettre.
- services génériques potentiellement attendus par les applications :
 - contrôle d'erreur
 - fiabilité
 - ordonnancement
 - contrôle de flux
 - contrôle de congestion

Couche Transport

- Deux principaux protocoles de la couche transport d'internet
 - TCP (*Transmission Control Protocol*)
 - procure une couche de transport fiable (alors même que IP ne l'est pas)
 - UDP (*User Datagram Protocol*)
 - ne fait que transporter de manière non fiable des datagrammes
- Tous les deux utilisent IP comme protocole de la couche réseau
 - Le champ protocole du datagramme IP est fixé à 6 pour TCP et à 17 pour UDP

Adressage des applications

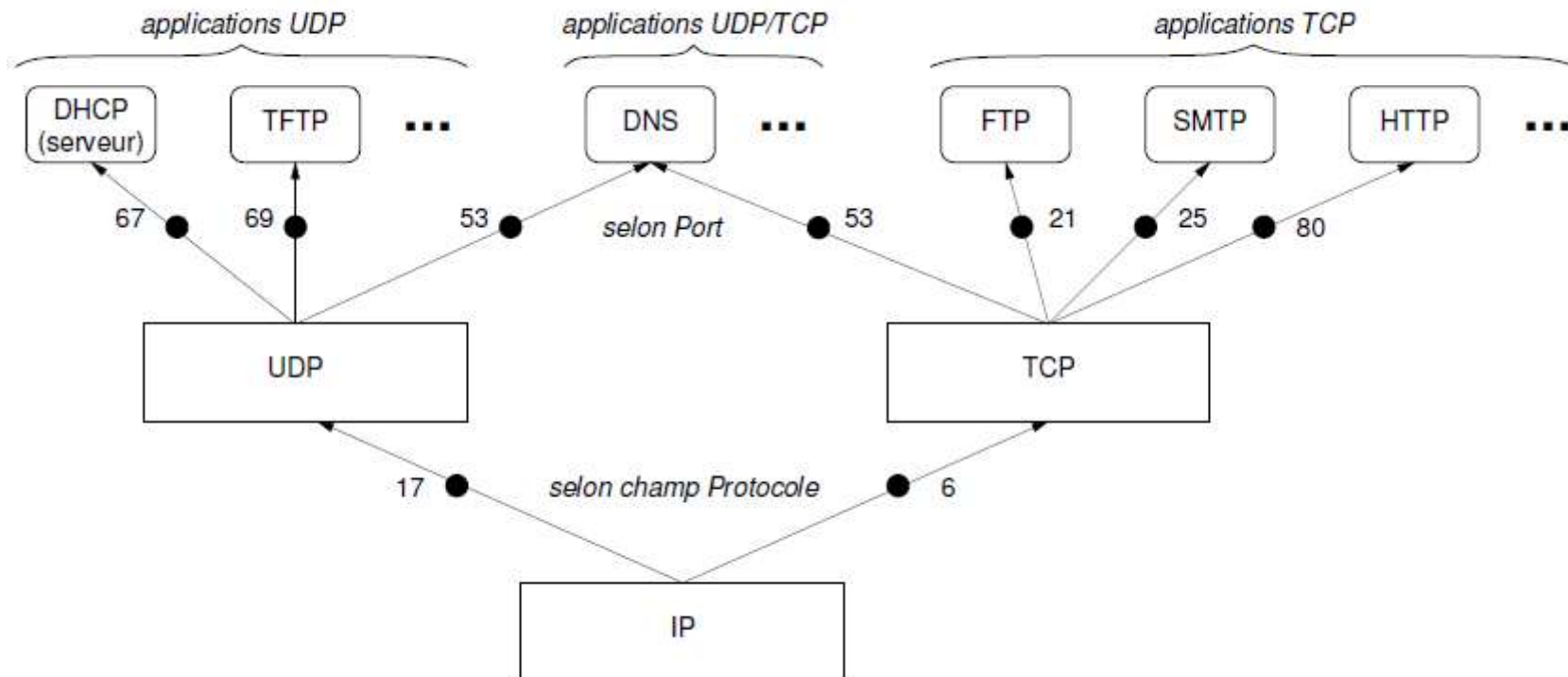
- Plusieurs applications réseaux peuvent s'exécuter en parallèle sur une machine
 - comment un émetteur peut-il préciser à quelle application est adressé un message ?
- La solution retenue sur Internet est l'utilisation de destinations abstraites : les ports
 - entiers positifs sur 16 bits
 - UDP et TCP fournissent chacun un ensemble de ports indépendants : le port n de UDP est indépendant du port n de TCP
 - le système permet aux applications de se voir affecter un port UDP et/ou TCP (choisi ou de manière arbitraire)
 - certains numéros de port sont réservés et correspondent à des services particuliers

Adressage des applications

- Parmi les 2^{16} numéros de ports disponibles, 3 zones sont définies
 - 0-1023 : ports reconnus, réservés aux services standards
 - 1024-49151 : ports enregistrés, utilisables temporairement par le système d'exploitation ou dans le domaine privé
 - 49152-65535 : ports publiques et dynamiques, libres d'utilisation
- L'adresse d'une application Internet est le triplet : (adresse IP, protocole de transport, numéro de port)
- L'association des identifiants (protocole, port destination, @IP destination, port source, @IP source) est désignée sous le terme de *socket* (RFC 793)

Multiplexage et démultiplexage

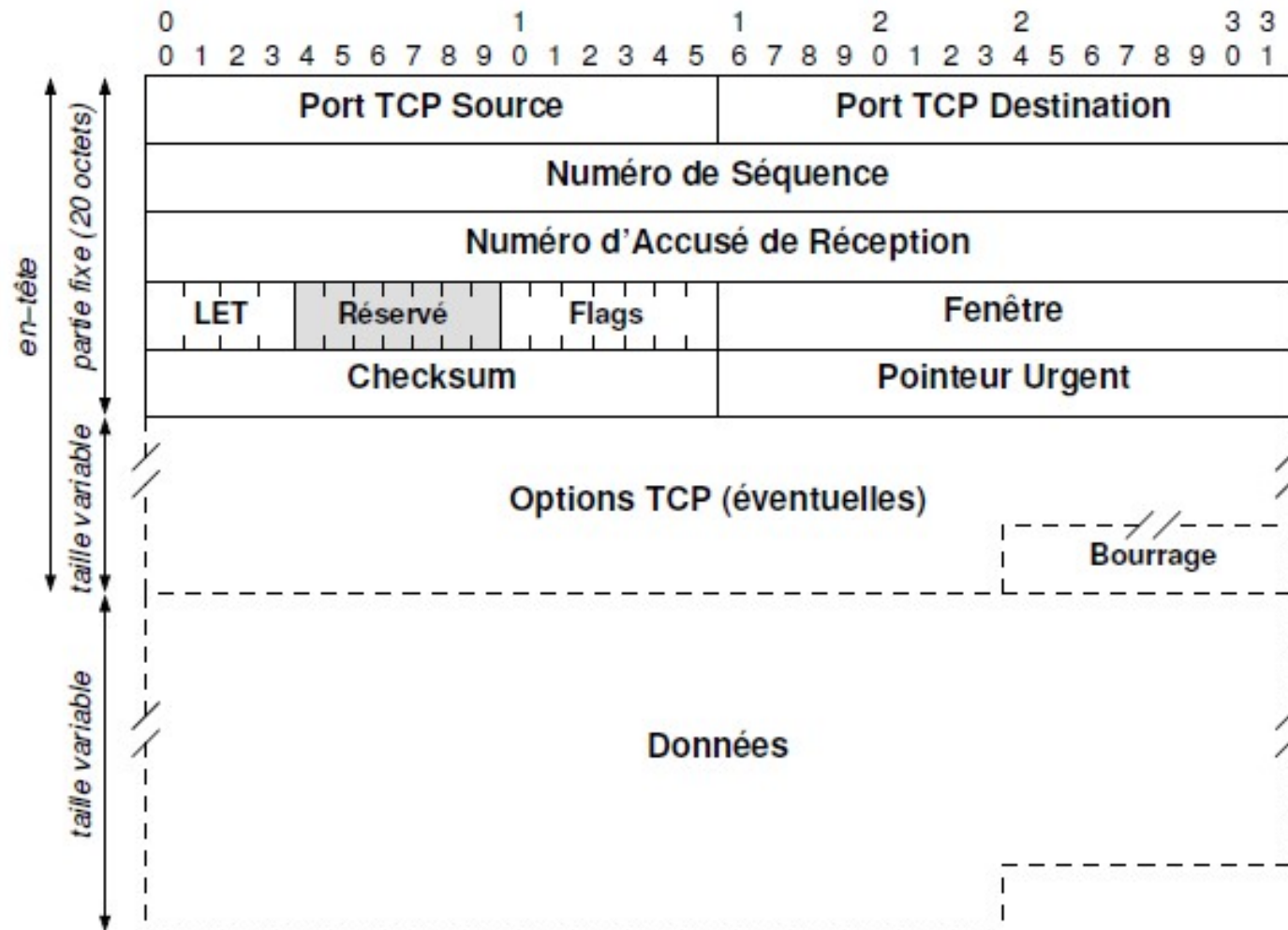
- La couche transport reçoit des données qu'elle doit acheminer sur le bon processus de l'application (connu par sa "socket")
- Les données émises sur plusieurs sockets doivent être agglomérées pour passer dans le tuyau (multiplexage)
- A l'arrivée, la distribution sur les sockets est le démultiplexage



Le protocole TCP

- Caractéristiques de base :
 - service fiable
 - connexion bidirectionnelle
 - mais un côté peut libérer un sens de transmission quand il n'a plus de données à émettre
 - point-à-point
 - ouverture en trois échanges
 - fermeture douce (négociée) ou brutale
 - orienté octet
 - connexions non structurées : pas de frontière placée par TCP entre les messages émis par les applications
 - taille maximum de segments (bloc d'octets)
 - L'option MSS (Maximum Segment Size)
 - MSS typiques : 1460, 1448 et 512

Format du segment TCP



Champs segment TCP

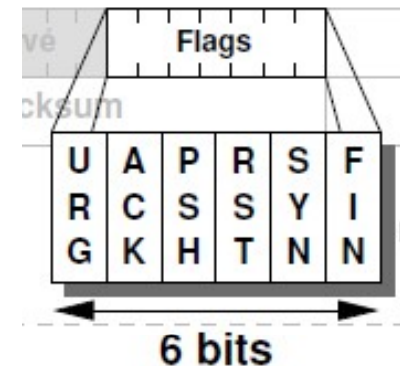
- Port TCP Source : indique le numéro de port de l'émetteur
 - Multiplexage à la source (plusieurs clients vers un serveur)
 - Peut être à 0 si aucune réponse n'est attendue
- Port TCP Destination : numéro de port du destinataire
 - Multiplexage à la destination (plusieurs serveurs sur un hôte)

Champs segment TCP

- Numéro de Séquence : Indique le numéro que porte le premier octet de données
 - associé à chaque octet et non pas à un segment
 - numérotation implicite des octets suivants
 - détection des pertes
 - ordonnancement
- Numéro d'Accusé de Réception : indique le numéro du prochain octet attendu
- Longueur de l'en-tête : nombre de lignes de 32 bits dans l'entête TCP
 - nécessaire car le champ option est de longueur variable
 - valeur de 5 (pas d'options) à 15 (10 lignes d'options, soit 40 octets)

Champs segment TCP

- Indicateurs (flags) : Chacun sur 1 bit
 - URG : présence de données urgentes
 - valide le contenu du champ « pointeur sur données urgentes »
 - ACK : champ acquittement valide
 - PSH : envoi immédiat avec vidage des tampons
 - RST : terminaison brutale de la connexion
 - SYN : synchronisation lors de l'ouverture
 - FIN : fermeture courtoise
- Fenêtre : indique la place disponible dans le tampon de réception
 - dimensionne la taille de la fenêtre d'anticipation (coulissante)
 - contrôle de flux
 - 16 bits (le récepteur annonce 64 Koctets maximum)
- Total de contrôle : permet de vérifier l'intégrité du segment
 - Obligatoire

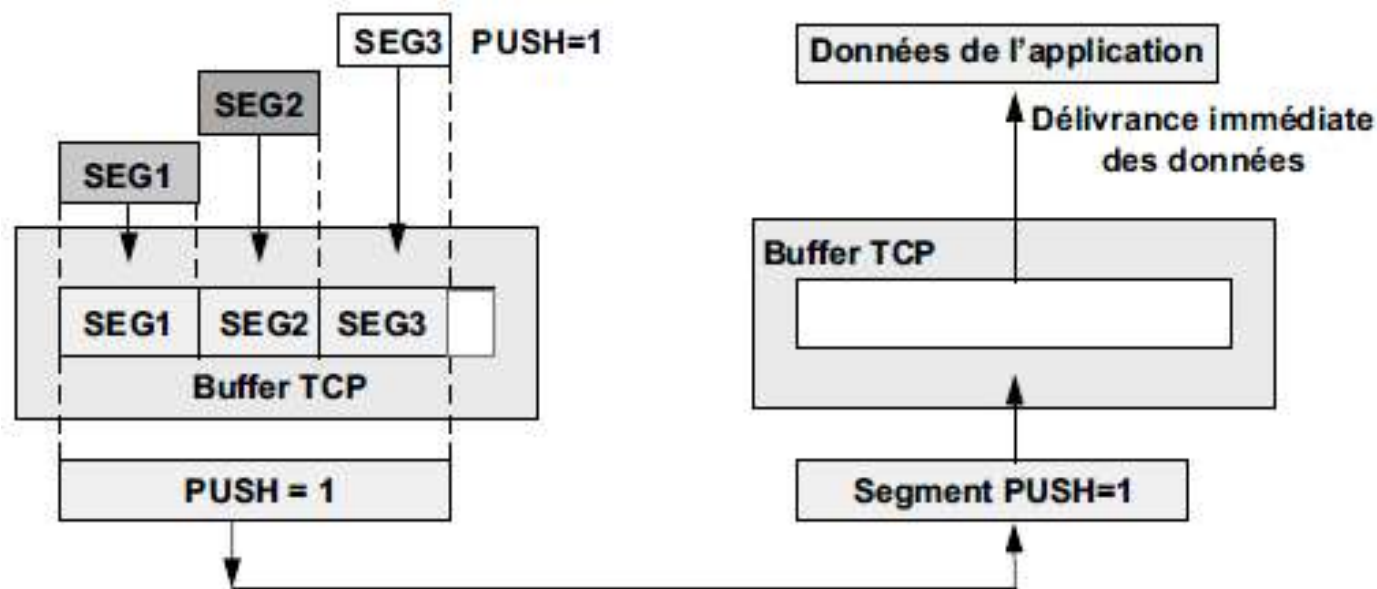


Segment TCP : champ Pointeur Urgent

- Emetteur veut envoyer des données en urgence, sans attendre que le récepteur ait lu les données précédentes
- Mise en œuvre par le bit URG et le champ Pointeur Urgent
- Si bit URG = 1 : données urgentes présentes et Pointeur Urgent valide
- Si bit URG = 0 : pas de données urgentes, Pointeur Urgent ignoré
- Pointeur Urgent : un offset positif qui, ajouté au numéro de séquence du segment, indique le numéro du dernier octet urgent du champ de données
 - les données comprises entre le début du champ données et la valeur du pointeur données urgentes sont traitées, en priorité, par le destinataire

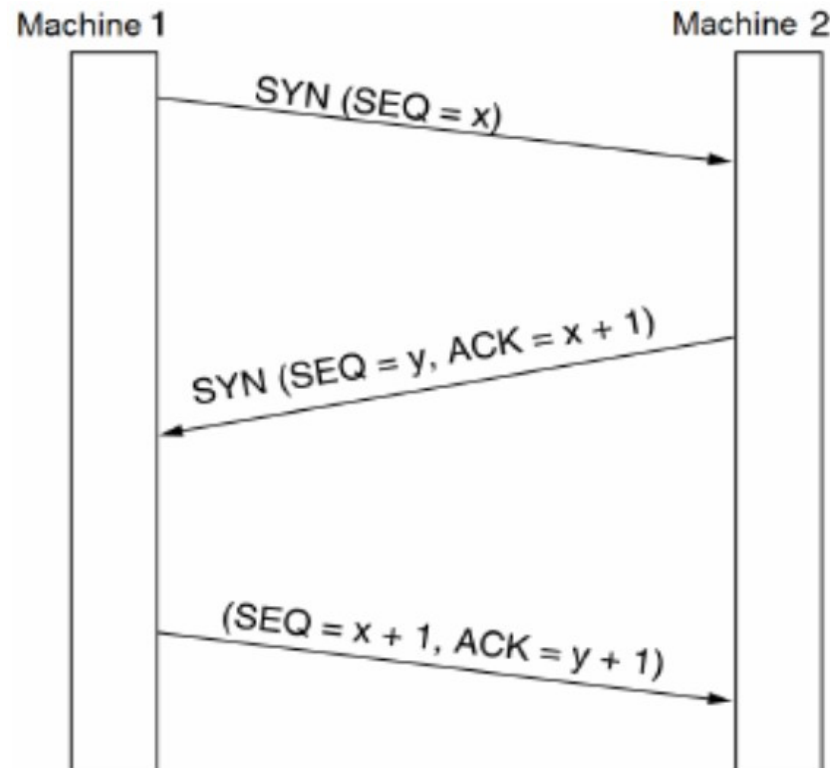
Bufferisation & Remise forcée

- Pour optimiser la transmission, TCP attend, avant de transmettre des données, que le *buffer* d'émission soit plein
- Certaines applications peuvent exiger la transmission immédiate
 - Particulièrement utile dans le cas d'un terminal virtuel
- La demande de positionnement du bit PUSH contraint le TCP émetteur à transmettre les données sans attendre
- Le TCP récepteur doit délivrer, immédiatement, les données reçues à l'application (ne pas tamponner dans sa mémoire)



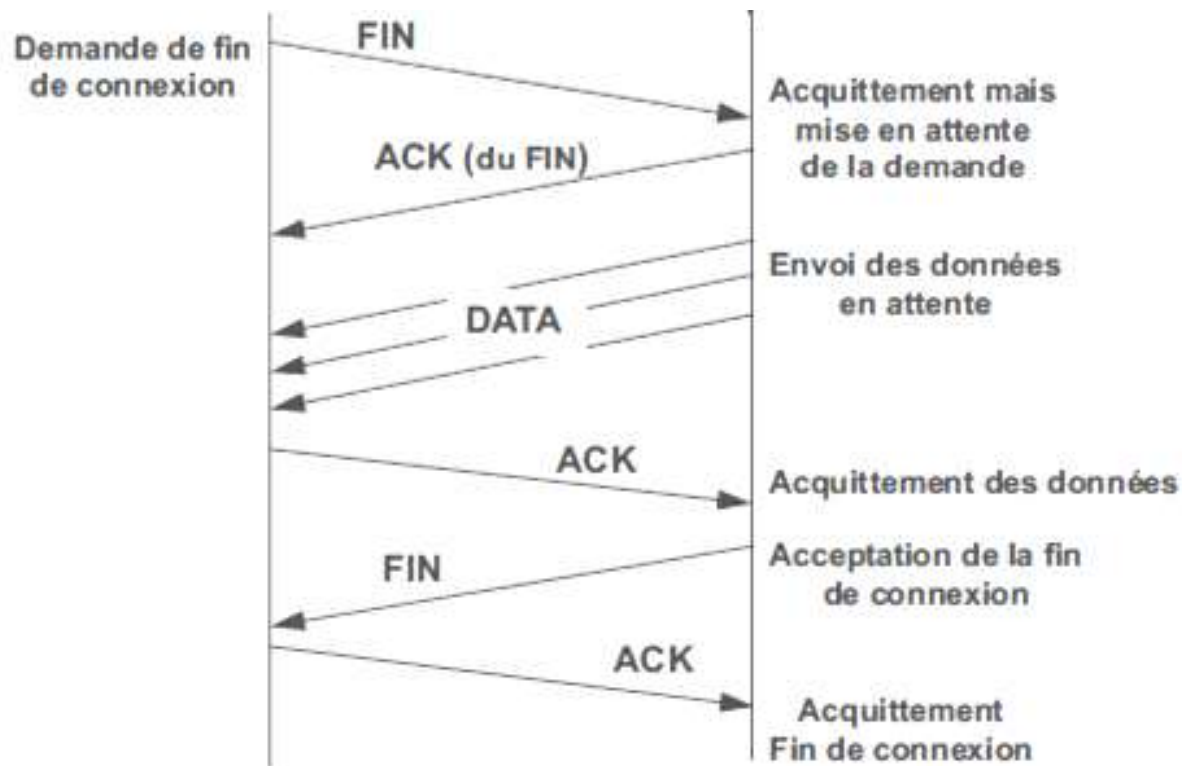
Etablissement d'une connexion

- En 3 temps, avec bits SYN et ACK
- Les numéros de séquence ne sont pas initialisés à zéro en début de transmission, mais à une valeur dite ISN (ISN, *Initial Sequence Number*)



Libération d'une connexion

- La connexion de transport est bidirectionnelle
- Il peut être mis fin à un échange dans un sens sans pour cela mettre fin à l'échange dans l'autre sens (semi fermeture)
- La fermeture définitive n'est effective que lorsque chaque sens a clos son échange



Fermeture brutale d'une connexion

- Bit RST a 1 :

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

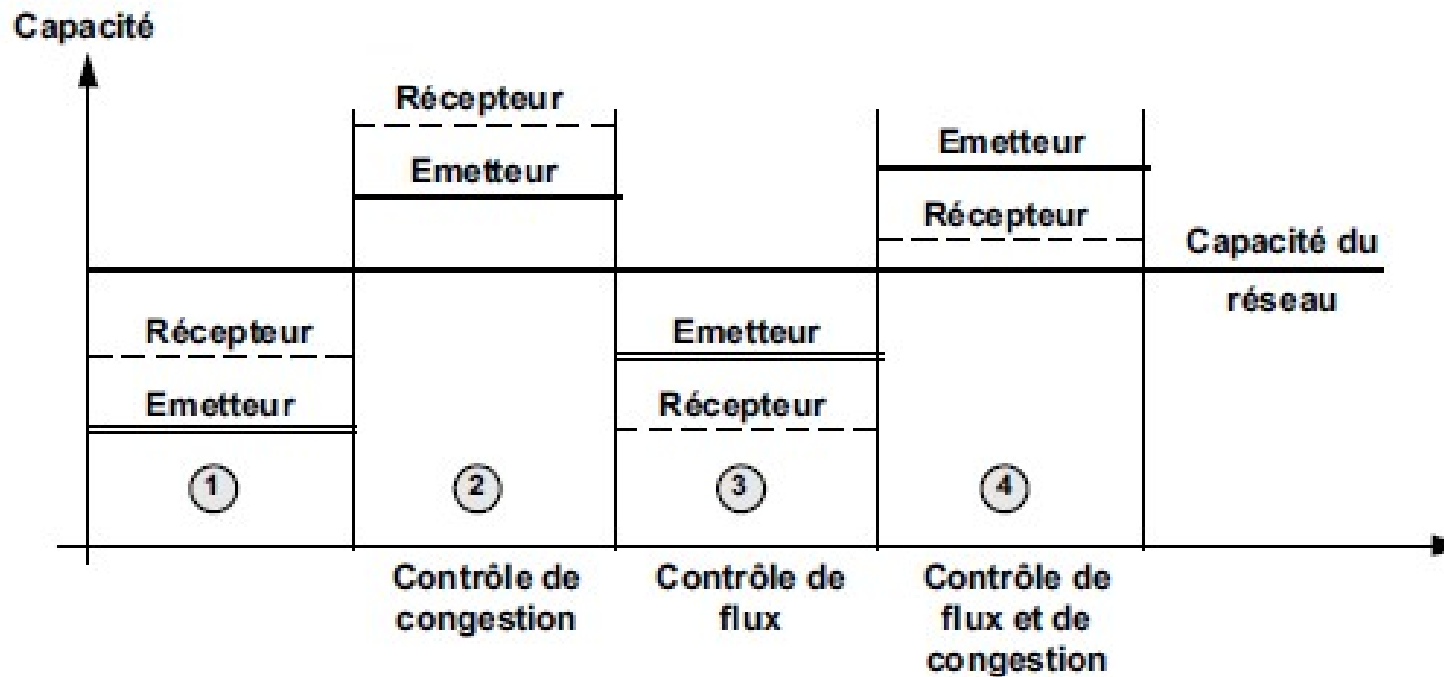
- Signifie qu'il a y eu un problème grave
- Connexion libérée immédiatement
- Données non traitées/segments retardés sont perdus
- Applications sont informées
- Utilise aussi pour refuser une demande de connexion

Le contrôle de flux et de congestion

- Le contrôle de flux consiste à contrôler le débit d'émission d'une source en fonction de la capacité de réception du destinataire
 - mécanisme de point à point entre deux entités communicantes
 - veille à ce qu'aucune donnée ne soit perdue par saturation des *buffers* du destinataire
- Le contrôle de congestion consiste à contrôler le débit d'émission d'une source en fonction de la capacité du sous-réseau physique réel
 - mécanisme de surveillance du trafic interne au réseau
 - s'assure qu'aucune donnée ne soit perdue par saturation des mémoires d'un quelconque nœud interne au réseau
- Dans les deux cas, de contrôler le débit de la source

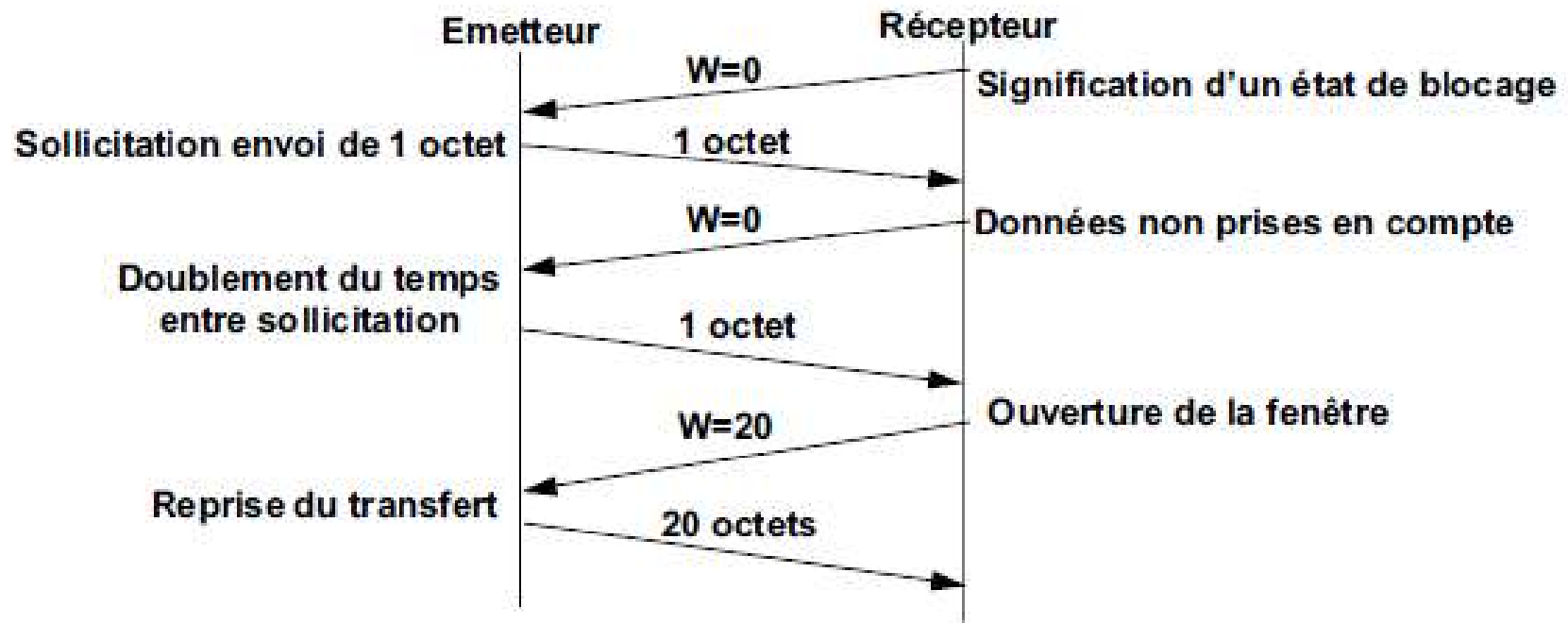
Le contrôle de flux et de congestion

- TCP gère deux fenêtres, une de contrôle de congestion et une de contrôle de flux
- Ces deux fenêtres sont indépendantes l'une de l'autre, le TCP émetteur limite ses émissions à la plus petite des deux valeurs

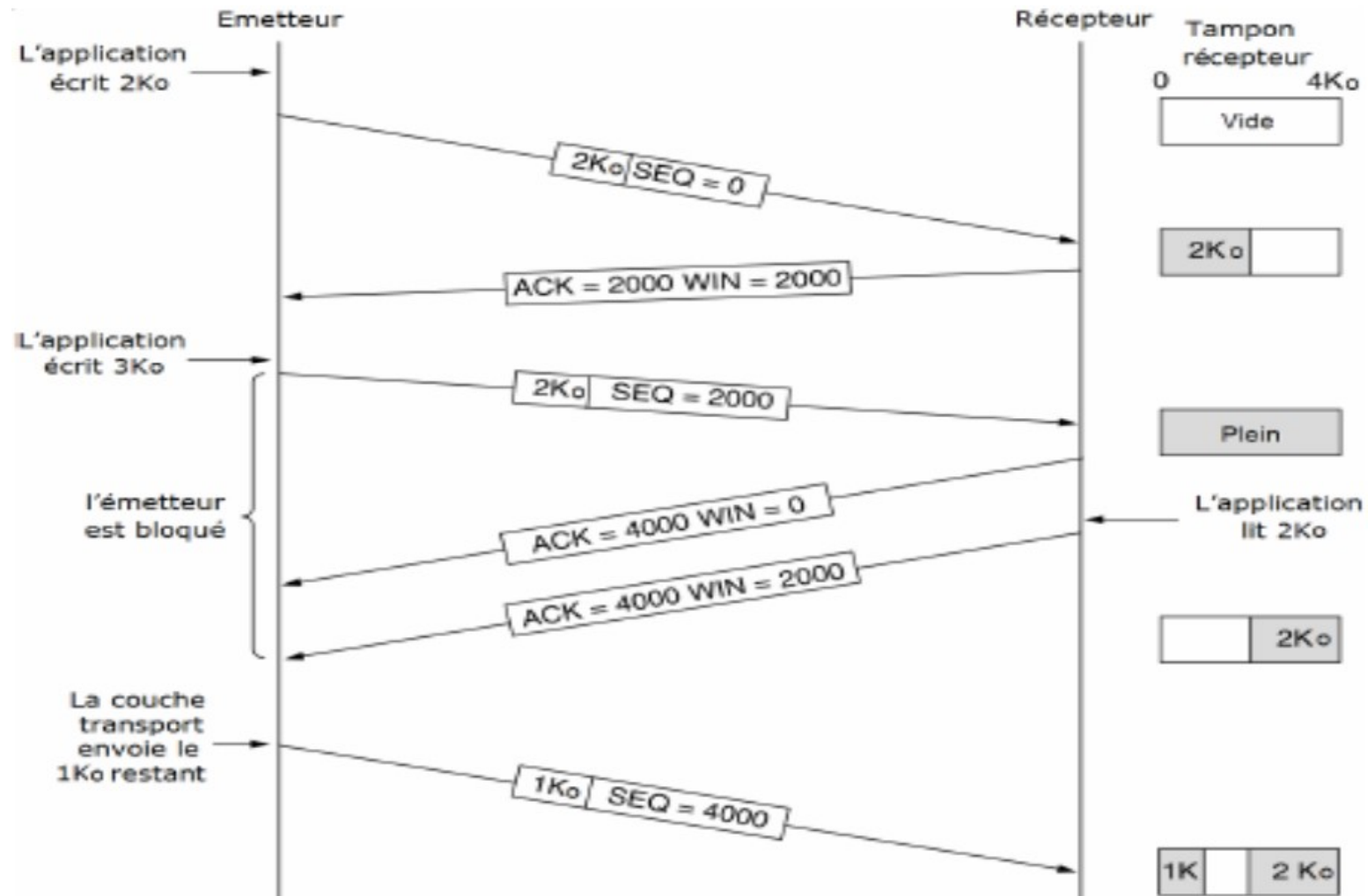


Le contrôle de flux

- Il s'agit d'un mécanisme à fenêtre dynamique
- Le crédit accordé est un crédit d'octets



Exemple

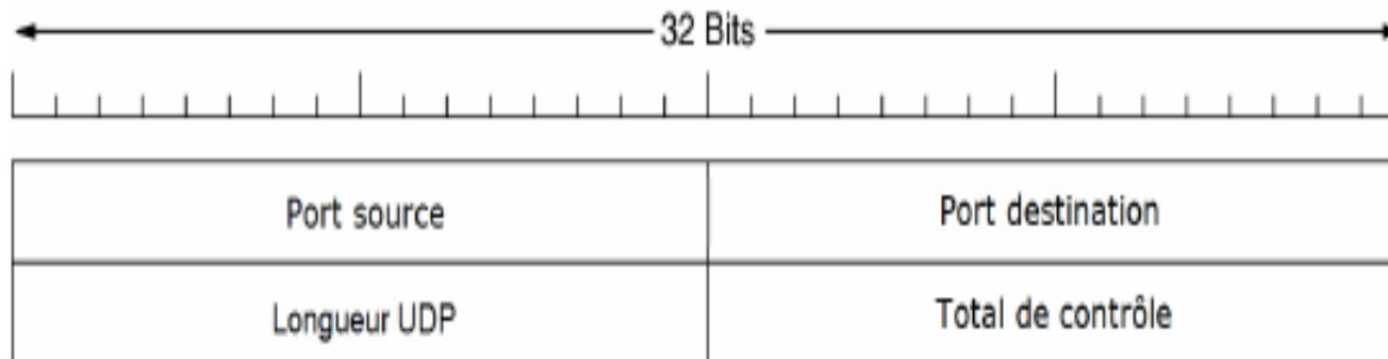


Le protocole UDP

- Le choix d'un service transport non connecté peut être motivé par :
 - pas d'établissement de connexion
 - absence d'état dans les hôtes
 - entête réduit
 - pas de retransmission
 - pas de contrôle du débit d'émission
- Service rendu :
 - envoi/réception de messages entre applications (processus) à travers Internet
 - adressage des applications par numéro de port
 - contrôle facultatif de l'intégrité des données
- Même type de service non fiable, non connecté que IP
 - possibilité de perte, duplication, déséquencelement de messages
 - pas de régulation de flux
- Un programme utilisant UDP doit gérer lui-même ces problèmes !

Format des datagrammes UDP

- En-tête : nombre fixe d'octets = 8
 - pas d'option possible
- Données : nombre variable d'octets ($\leq 65\,535$)



Champs UDP

- Port UDP Source : indique le numéro de port de l'émetteur
 - Multiplexage à la source (plusieurs clients vers un serveur)
 - Peut être à 0 si aucune réponse n'est attendue
- Port UDP Destination : numéro de port du destinataire
 - Multiplexage à la destination (plusieurs serveurs sur un hôte)
 - le serveur doit être en écoute sur ce port pour accepter le datagramme
- Longueur totale avec les données exprimée en octets
 - 16 bits (64 KOctets maximum)
- Total de contrôle : permet de vérifier l'intégrité du datagramme
 - Facultatif (la valeur 0 s'il n'est pas calculé)

Files d'attente et réception

- Attribution de port → allocation de file d'attente
- Réception d'un datagramme :
 - port destination non attribué : destruction + message ICMP
 - port alloué : si file non pleine, ajouté dans la file, sinon détruit (pas de message ICMP!)
- Appli consomme les éléments de la file, par accès synchrone :
 - si aucun datagramme, processus placé en attente
 - sinon, consommation du premier datagramme (FIFO)
- Une application peut avoir plusieurs ports

Utilisation UDP : multidiffusion

- UDP offre un service en mode non connecté
- Une application peut ainsi exploiter la multidiffusion IP et émettre un seul datagramme qui sera reçu par un ensemble de stations, en utilisant comme destination, une adresse :
 - broadcast : diffusion dirigée à toutes les stations d'un réseau, ou du réseau local pour l'adresse de diffusion limitée (255.255.255.255)
 - multicast : adresses de classe D pour diffuser à un ensemble de stations se trouvant éventuellement sur des réseaux différents
 - Utilisées notamment pour la diffusion de medias audio/vidéo.
- Exemple : envoi par un client DHCP d'un message à tous les serveurs DHCP du réseau local. L'adresse destination du message DHCP est (255.255.255.255, UDP, 67)