La couche transport

Sommaire

- Couche transport
- TCP : Transmission Control Protocol
- Send and Wait
- Mécanisme de fenêtrage
- Acquittement dans TCP
- Retransmission dans TCP
- UDP : User Datagram Protocol

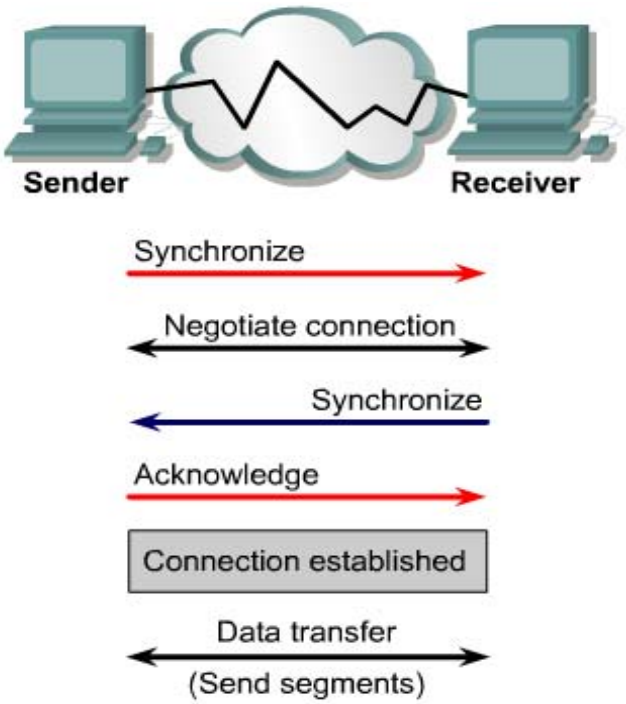
Couche transport

- Fonctions :
  - Assurer la transmission de bout en bout
  - Remédier aux imperfections des couches plus basses
    - pertes, erreurs, paquets en désordre
  - Contrôle de flux (ou de trafic)
    - Ensemble d'actions permettant de prévenir la congestion.
    - Ajuster les vitesses d'envoi et de réception
  - Traiter des cas de congestion dans le réseau
    - État où l'ensemble des ressources requises par les sources excèdent ceux du réseau.
    - Conséquences : dégradation des performances (taux de perte, délais des paquets)
  - Optimiser les performances du transfert

Couche transport

- Le protocole IP permet de mettre en correspondance 2 machines grâce à leurs adresses IP; Mécanisme incomplet puisque la communication s'effectue au niveau des processus qui tournent sur ces machines.
- Nécessité d'identifier quels sont les processus des machines qui veulent communiquer.
- Adressage (niveau couche transport) basé sur la notion de numéro de port :
  - Socket ↔ couple <@IP, port>
  - Connexion ↔ double couple <@IP, port> émetteur et récepteur
  - Multiplexage ↔ plusieurs connexions utilisant la même adresse réseau
- Deux protocoles de transport :
  - TCP : Transmission Control Protocol
  - UDP : User Datagram Protocol

Principe d'un échange TCP

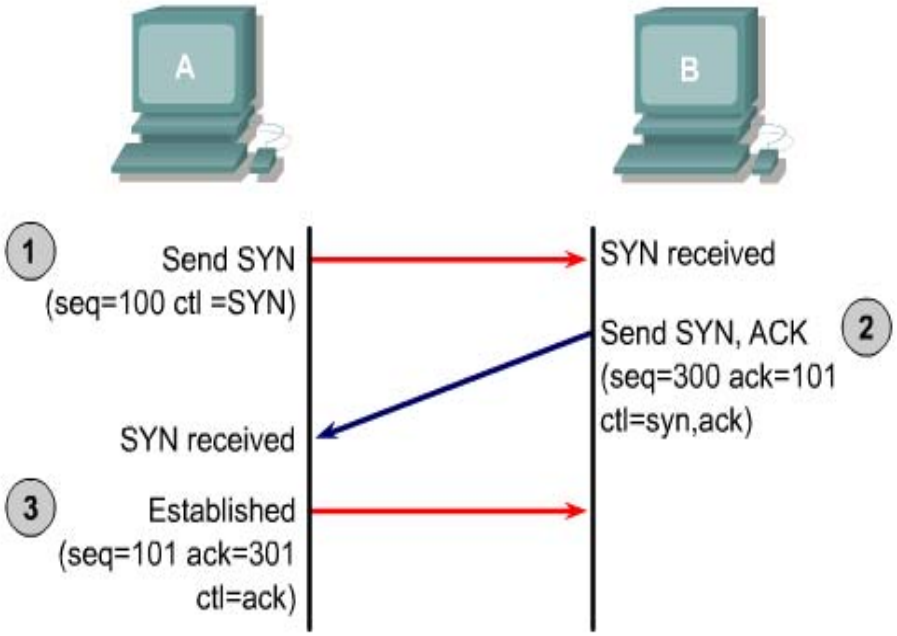


Source: Cisco system

Principe d'un échange TCP

- L'établissement de la connexion :
  - par triple échange ("three-way handshake")
- Une machine A initialise la connexion en envoyant un segment incluant un **SYN** (*Synchronize sequence numbers)* et un **numéro de séquence x**.
- La machine B lui répond par un segment avec les drapeaux **SYN** et **ACK** (*Acknowledgement*) avec un **numéro d'acquittement x+1**. Son **numéro de séquence vaut y**.
- La machine A renvoie alors un segment avec le flag **ACK** et le **numéro d'acquittement y+1**.

Three-Way Handshake



Source: Cisco system

TCP : segmentation

- Segmentation,
  - Traite les données venant des couches supérieures comme un flot d'octets.
  - Découpe ce flot d'octets en segments dont la taille est fonction du mécanisme de fenêtrage.
  - Un segment TCP est émis dans un datagramme IP.

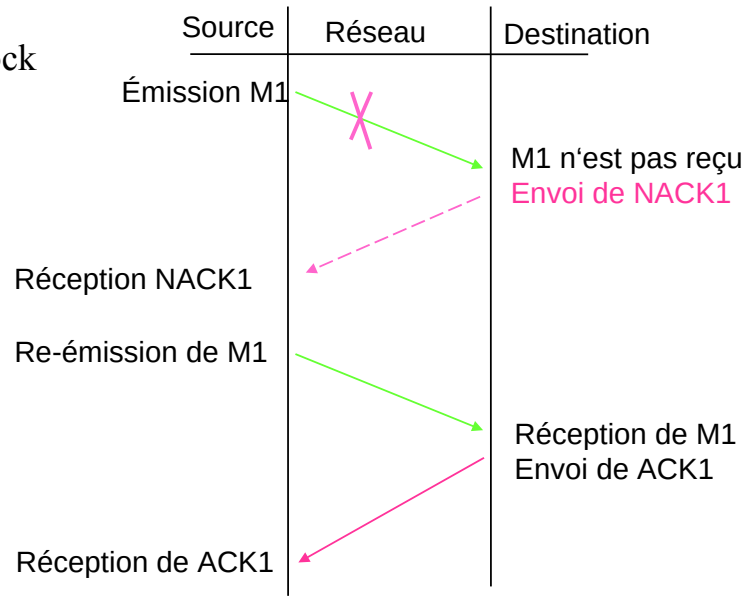
- Acquittement de messages
  - TCP garantit l'arrivée des messages, c'est à dire qu'en cas de perte, les deux extrémités sont prévenues.

TCP : Transfert de données

- Le protocole TCP comptabilise les octets transmis.
  - Le champ **seq** indique la place du premier octet de données du paquet ;
  - Le champ **ack** indique le prochain octet attendu par l'émetteur du paquet;

Send and Wait (1)

- Acquittement de messages On transmet un segment puis en attend l'acquittement avant de transmettre le suivant :
  - Fiable transmission,
  - sender can send the next block after receiving ACK
  - without loose

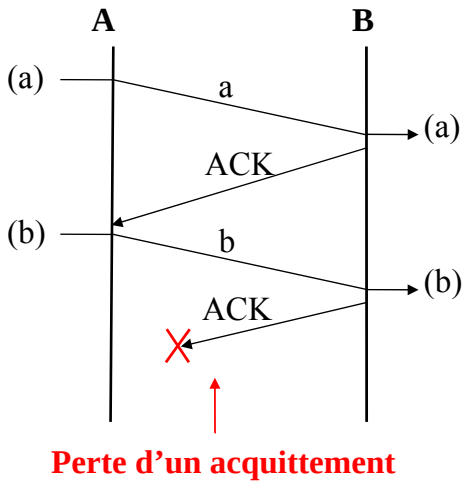


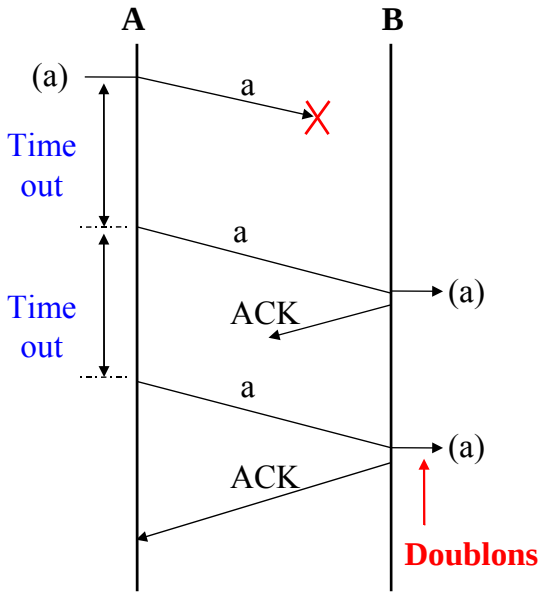
Send and Wait (2)

- Temporisateur armé lors de l'émission :
  - si pas d'acquittement à échéance du temporisateur, on retransmet et on attend à nouveau …
  - moins efficace (mais plus sûr) que par acquittement explicite (négatif) en provenance du récepteur.
  - risque de duplication de paquets lors de la retransmission.

Send and Wait (2)

- Problème n°1 : Perte d 'une trame ou d 'un ACK

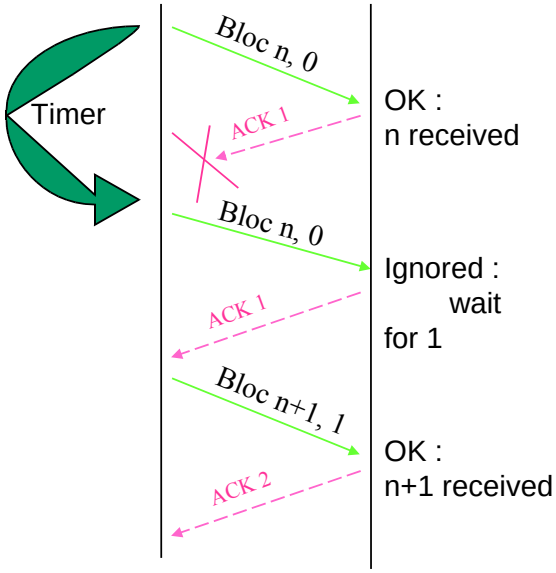




- Solution : associer les acquittements aux paquets

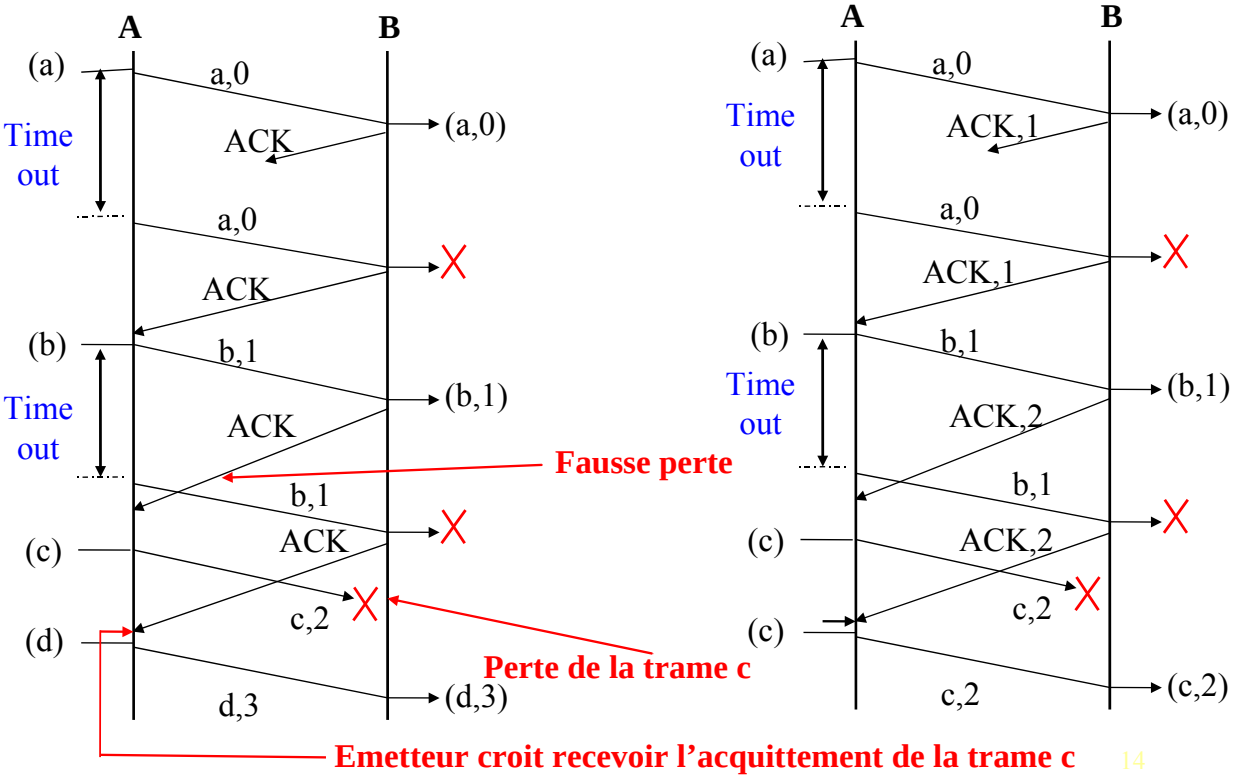
Send and Wait (3)

- Numérotation des paquets et des acquittements :
  - Compteur à k bits mod. 2^k
    - n° SEQ de 3-bit, l'espace est 0,…,7
  - Très utile pour le reséquencement
  - Deux considérations importantes :
    - choosing sequence number length
    - choosing initial sequence number



Send and Wait (3)

- Problème n°2 : Doublons

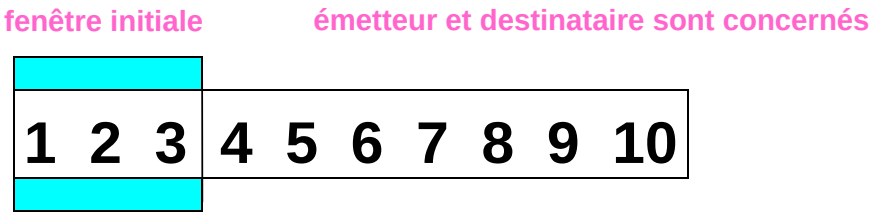


Le fenêtrage

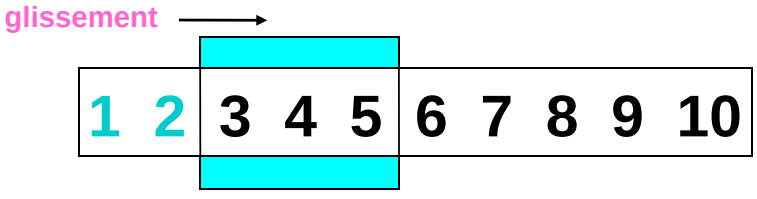
- La technique acquittement simple pénalise les performances:
  - Il faut attendre un acquittement avant d'émettre un nouveau message.
- Introduction de la notion de Fenêtre d'Anticipation … (fenêtre de transmission) pour améliore le rendement du réseau.
- La technique du fenêtrage
  - une fenêtre de taille T, permet l'émission d'au plus T messages "non acquittés" avant de ne plus pouvoir émettre.

Technique de fenêtrage (1)

- Mécanisme de glissement de fenêtre (sliding window)

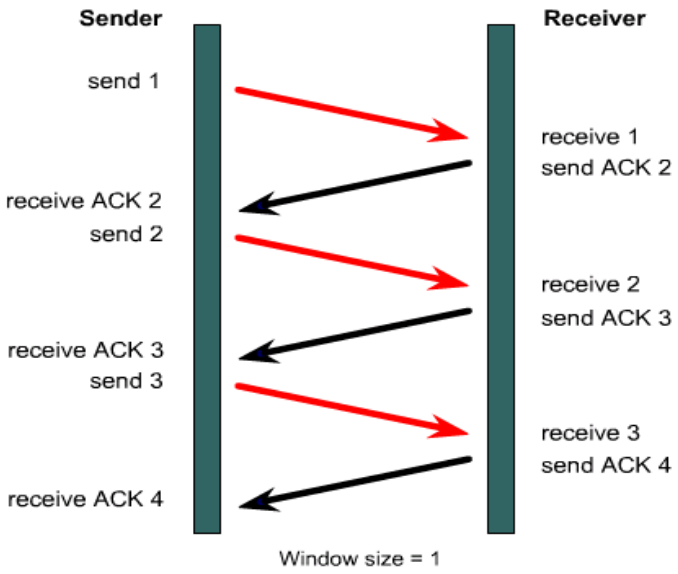


  - L'émetteur peut envoyer 3 paquets avant de recevoir un acquittement
  - L'acquittement du paquet 2 arrive, la fenêtre glisse



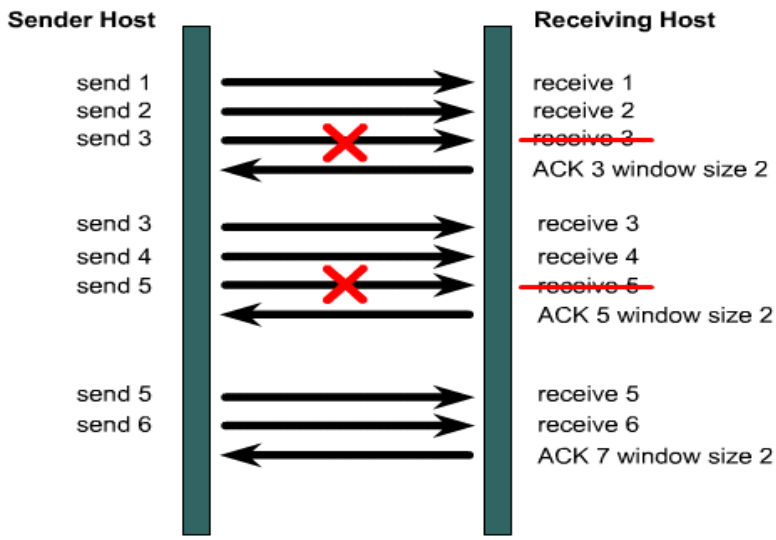
  - Les performances sont fonctions de la taille de la fenêtre et de la vitesse à laquelle le réseau accepte les paquets.

TCP fenêtrage Basique



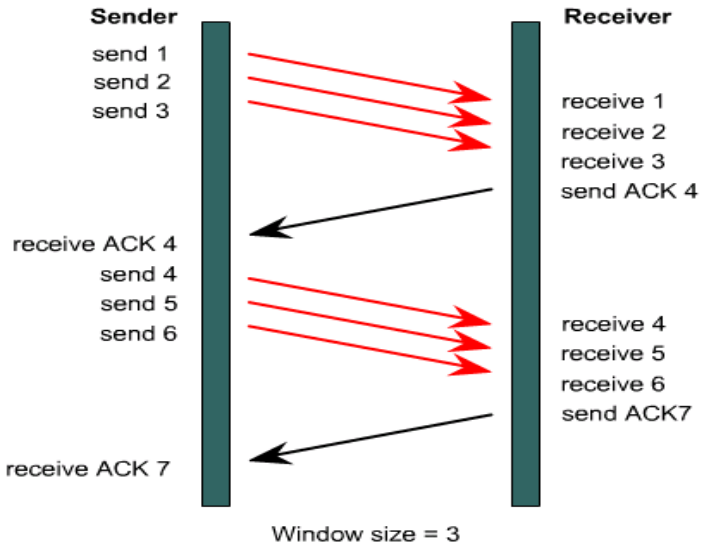
Source: Cisco system

TCP Sliding Window



Source: Cisco system

TCP Sliding Window

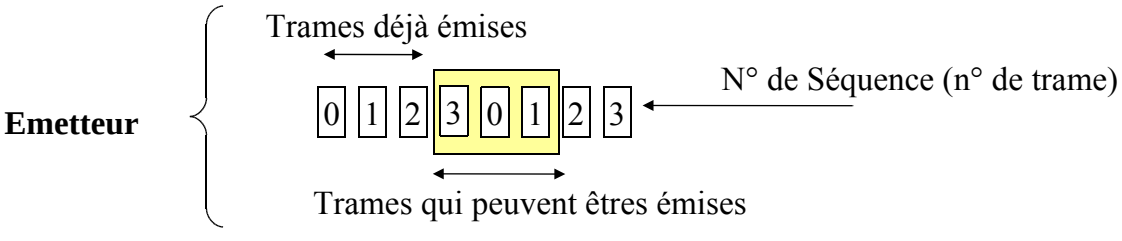


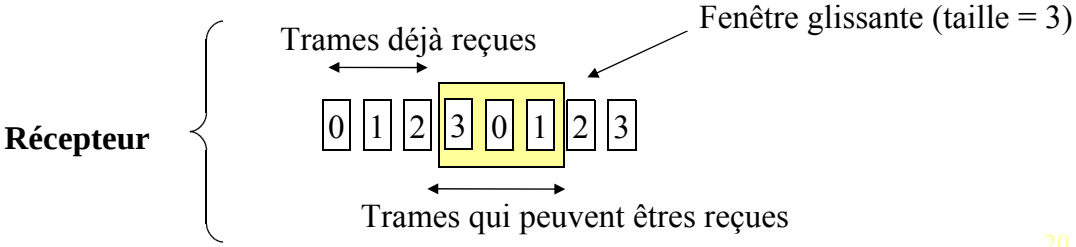
Source: Cisco system

Technique de fenêtrage (2)

Principe :

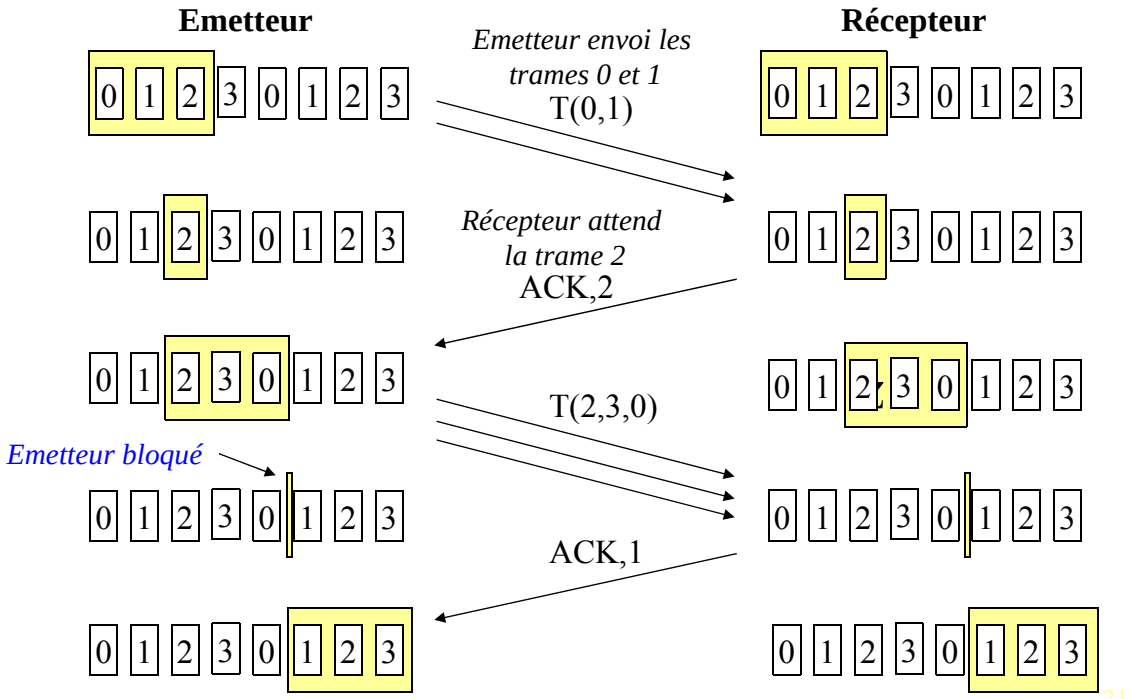
- plusieurs trames émises à la suite
- Trames numérotées sur « n » bits et modulo 2^n
- Taille des fenêtres : 2^n-1





Technique de fenêtrage (3)

(Numérotation sur 2 bits)



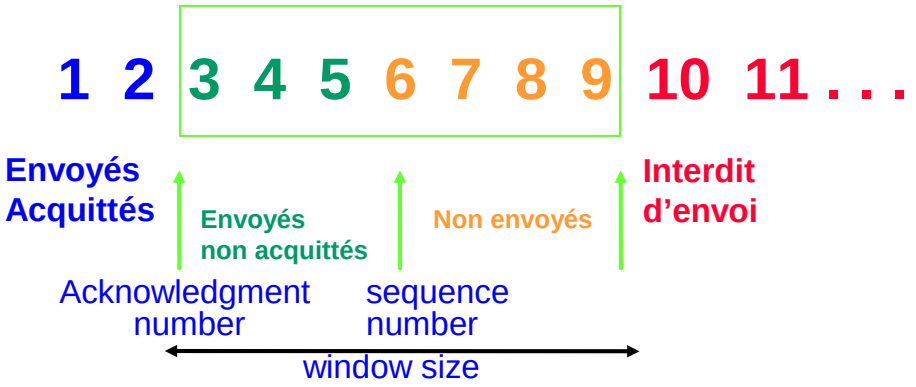
Technique de fenêtrage (4)

- Le **sliding window** permet également au destinataire de faire diminuer le débit de l'émetteur, donc de gérer le contrôle de flux, par la diminution de la taille de la fenêtre.

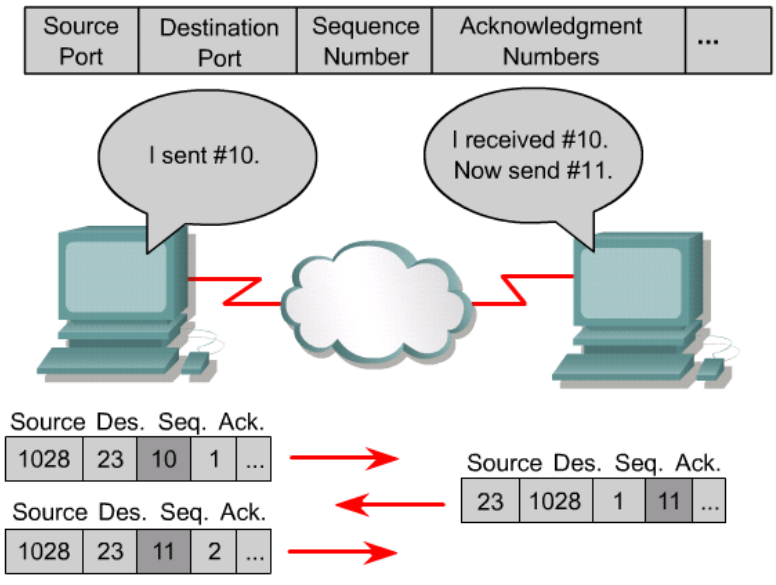
- Dans TCP, le mécanisme de fenêtrage opère au **niveau de l'octet** et non pas au niveau du segment; il repose sur :
  - la numérotation séquentielle des octets de données,
  - la gestion de trois pointeurs par fenêtrage :

Technique de fenêtrage (5)

- Trois pointeurs :
  - **Sequence number** : Numéro de premier octet du champ de données dans le flux d'octets transmis ;
  - **Acknowledgment number** : Numéro du prochain octet à recevoir. Acquitte tous les octets de numéro inférieur ;
  - **Window size** : Nombre d'octets pouvant être envoyés par anticipation (sans recevoir d'acquittement). Capacité de stockage du récepteur.



TCP Sequence and Acknowledgement



Source: Cisco system

TCP : Acquittements

- Le mécanisme d'acquittement de TCP est cumulatif
  - Il indique le numéro de séquence du prochain octet attendu.
    (tous les octets précédents cumulés sont implicitement acquittés)
  - Si un segment a un numéro de séquence supérieur au numéro de séquence attendu (bien que dans la fenêtre), le segment est conservé mais l'acquittement référence toujours le numéro de séquence attendu.
    - Permet de détecter des pertes au niveau récepteur
  - La récupération des pertes se fait par retransmission.

TCP : Retransmission

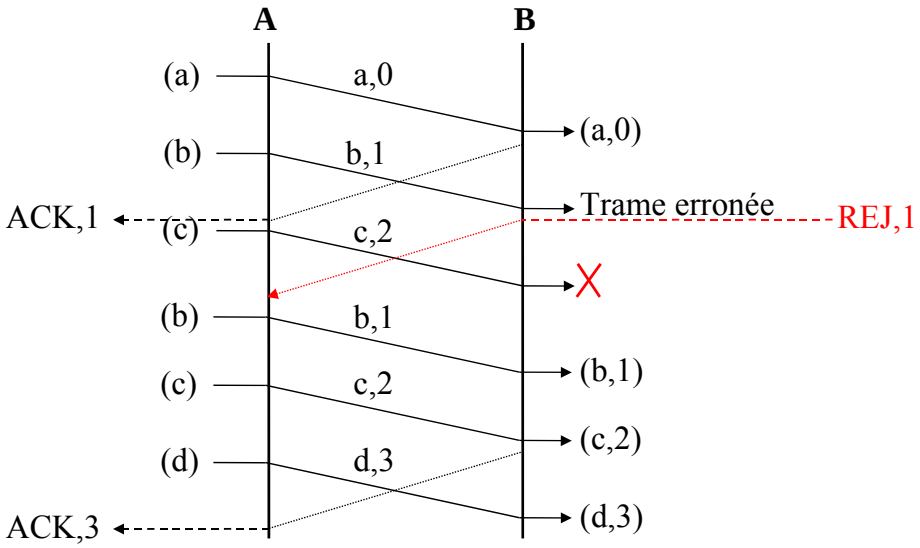
- Détection des pertes :
  - A l'émetteur : par l'expiration du temporisateur, par réception d'acquittements dupliquées.
- Récupération des pertes par retransmission :
  - Go-back-N : retransmission à partir du 1er segment non acquitté;
    - Utilisé dans TCP : TCP Reno (90), TCP Tahoe (88)
    - No buffer at receiver
    - Require buffer at sender
  - Inconvénient de la stratégie : Les paquets retransmis peuvent être déjà reçus par le récepteur :
  - Solution : Utiliser des acquittements sélectifs

TCP : Go Back N

« GO-BACK-N » :

- La transmission est reprise depuis la trame perdue ou erronée

  Méthode simple mais peu efficace



TCP : Retransmission

- Récupération des pertes par retransmission :
  - Selective Repeat : Retransmettre uniquement les segments perdus ;
    - re-sequencing buffer at receiver
    - Option SACK introduite dans la RFC 2018 (TCP SACK, 99)
  - Principe : Reporter dans les acquittements les blocs contigus de segments reçus par le récepteur :
    - début et fin de chaque bloc de segments contigus reçus

Selective Repeat

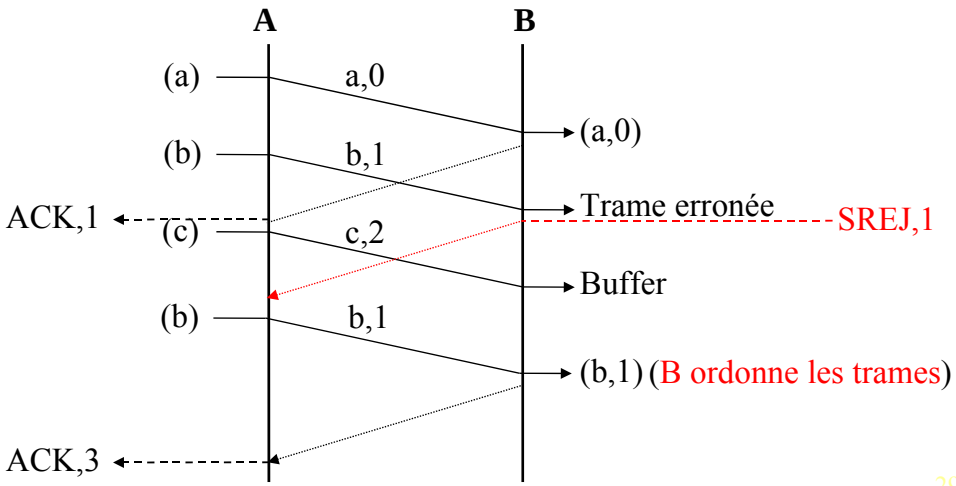
«Selective Repeat» :

- Seule la trame perdue ou erronée est retransmise

  Plus efficace que le GO-BACK-N

  Implémentation plus complexe

  Mémorisation des trames côté récepteur



TCP : les horloges

- RTT (Round Trip Time)
  - Temps entre l'envoi d'un paquet et la réception de son accusé
- RTO (Retransmission Timeout)
  - A chaque envoi d'un paquet, une horloge propre est lancée
  - si l'horloge expire, le paquet est retransmis
    - le RTO est affecté dynamiquement, en fonction du RTT [RFC 2988]
    - Par exemple : RTO = 2*RTT

TCP : les fenêtres

- Fenêtre d'émission : les données de l'application
- Fenêtre de réception (rwnd)
  - Nombre max de paquets que le récepteur est capable de recevoir = taille du buffer en entrée du récepteur
- Fenêtre de congestion (cwnd)
  - Sous-fenêtre mobile de la fenêtre d'émission
  - Nombre max de paquets que l'émetteur peut envoyer avant de recevoir un accusé lui permettant de poursuivre la transmission
  - Représente la quantité maximale de données en transit dans le réseau
- Seuil de démarrage lent (ssthresh)
  - Estimation de la bande passante disponible, généralement initialisé à la valeur de rwnd

TCP : Congestion

- Contrôle de flux (CF):
  - L'émetteur adapte le nombre de paquets envoyés à la taille du buffer de réception.
  - Flow control is for receivers
- Contrôle de congestion (CC):
  - L'émetteur adapte le débit des données envoyées à la bande passante instantanée du réseau.
  - Congestion control is for the network
  - Attention ce n'est pas la taille des paquets qui change, mais leur débit d'envoi.

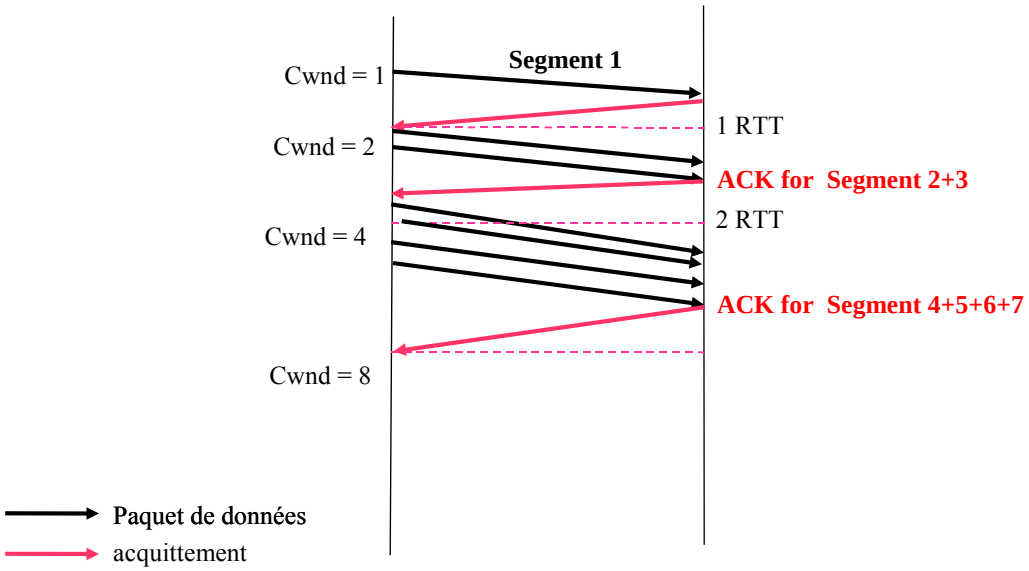
TCP : contrôle de congestion

- Mécanismes CC:
  - En gén., la perte d'un paquet est la seule information sur l'état du réseau
  - Le CC est géré exclusivement par l'émetteur
    - le récepteur ne fait que renvoyer des accusés de réception
  - Les mécanismes basiques de CC supportés par TCP sont [RFC 2581] :
    - slow start
    - congestion avoidance
    - …

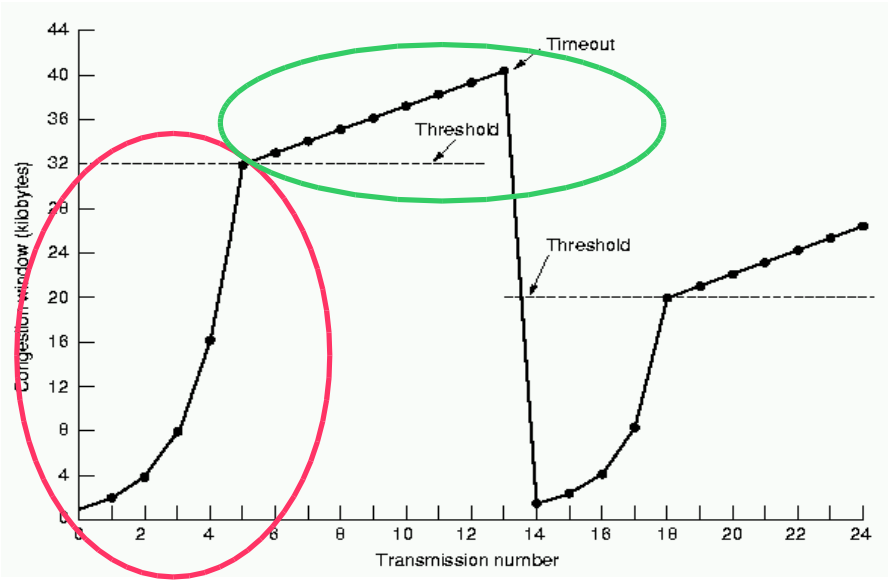
TCP : Slow Start

- But : retrouver rapidement la bande passante disponible
  - cwnd = 1, à l'établissement d'une connexion
  - ssthresh = valeur arbitraire
  - Augmentation progressive de cwnd (au rythme des RTT ; à chaque accusé reçu) jusqu'à détection d'une perte
    - cwnd = cwnd * 2 (croissance exponentielle)
- Si timeout (perte d'un paquet):
  - Réduction de la taille de cwnd et on relance le slow start
    - cwnd = 1
  - ssthresh = cwnd / 2

Slow Start



TCP congestion control: the big picture



→ cwnd grows exponentially (slow start), then linearly (congestion avoidance) with 1 more segment per RTT

→ If loss, divides threshold by 2 (multiplicative decrease) and restart with cwnd=1 packet

TCP : Congestion Avoidance

- Utilisé quand cwnd >= ssthresh

- But : augmenter le débit en testant « ***gentillement*** » la bande passante disponible
  - Augmenter cwnd à chaque fois qu'une fenêtre de congestion entière est acquittée
  - Croissance linéaire pour éviter de revenir trop rapidement à une phase de congestion

- Si timeout :
  - ssthresh = cwnd / 2
  - cwnd = 1
  - Retour au mode slow start

TCP : Améliorations

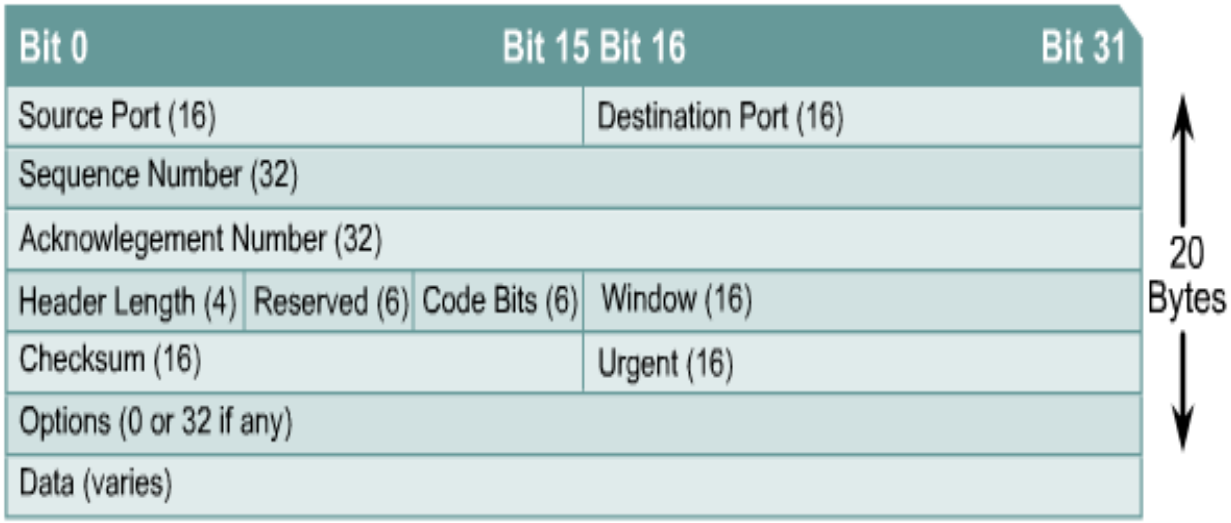
- Plusieurs constats :
  - Le passage à cwnd = 1, à chaque perte de paquet gaspille la bande passante
  - Une détection plus rapide des pertes de paquets permet d'avoir un débit plus élevé

- Deux nouveaux mécanismes :
  - fast retransmission: Utilisation d'acquittements dupliqués
  - fast recovery: passage à la phase de congestion avoidance dès la réception du 3eme paquet dupliqué.

Transmission Control Protocol

- Transmission de données :
  - Paquets de tailles variables (champ de données de longueur variable)
  - En mode connecté (avec phase de connexion)
- Fiabilité :
  - Sans erreur : contrôle et retransmission si nécessaire;
  - Sans perte : numérotation et retransmission;
  - Système d'acquittement positive;
  - Reséquencement des paquets.
- Contrôle de flux et de congestion :
  - Fenêtre d'anticipation modulé par le récepteur
  - Adaptation à l'état d'occupation du réseau

TCP Segment Format



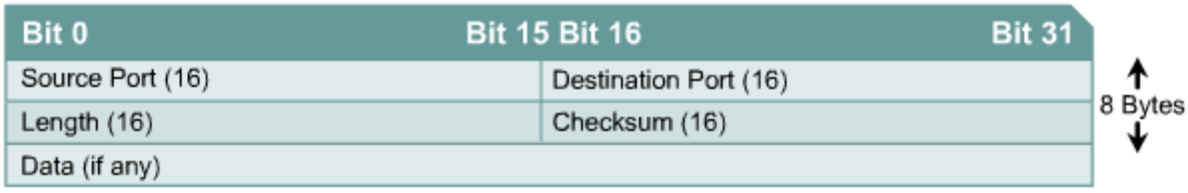
Source: Cisco system

UDP : User Datagram Protocol

- UDP : protocole de transport sans connexion :
  - émission de messages sans établissement de connexion au préalable.
  - l'arrivée des messages ainsi que l'ordonnancement ne sont pas garantis.

UDP : format des messages

Les messages UDP sont également appelés des datagrammes UDP.
Ils contiennent deux parties : un en-tête UDP et les données UDP.

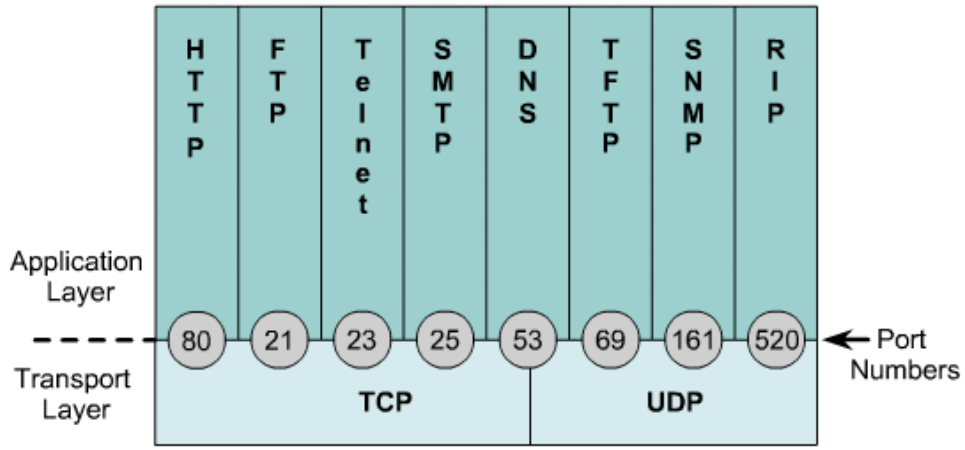


Format des messages UDP

Les ports source et destination contiennent les numéros de port utilisés par UDP.

La longueur du message est exprimée en octets (8 au minimum) (en-tête + données), le champ de contrôle est optionnel (0 si non utilisé).

TCP & UDP: ports standards



Références

- Central Web (Site intéressant)
- Réseaux et Télécoms, édition Dunod (2003), C. Servin
- Practical Unix and Internet Security, O'REILLY Edition
- Transparents *Walid Dabbous (*INRIA Sophia Antipolis)