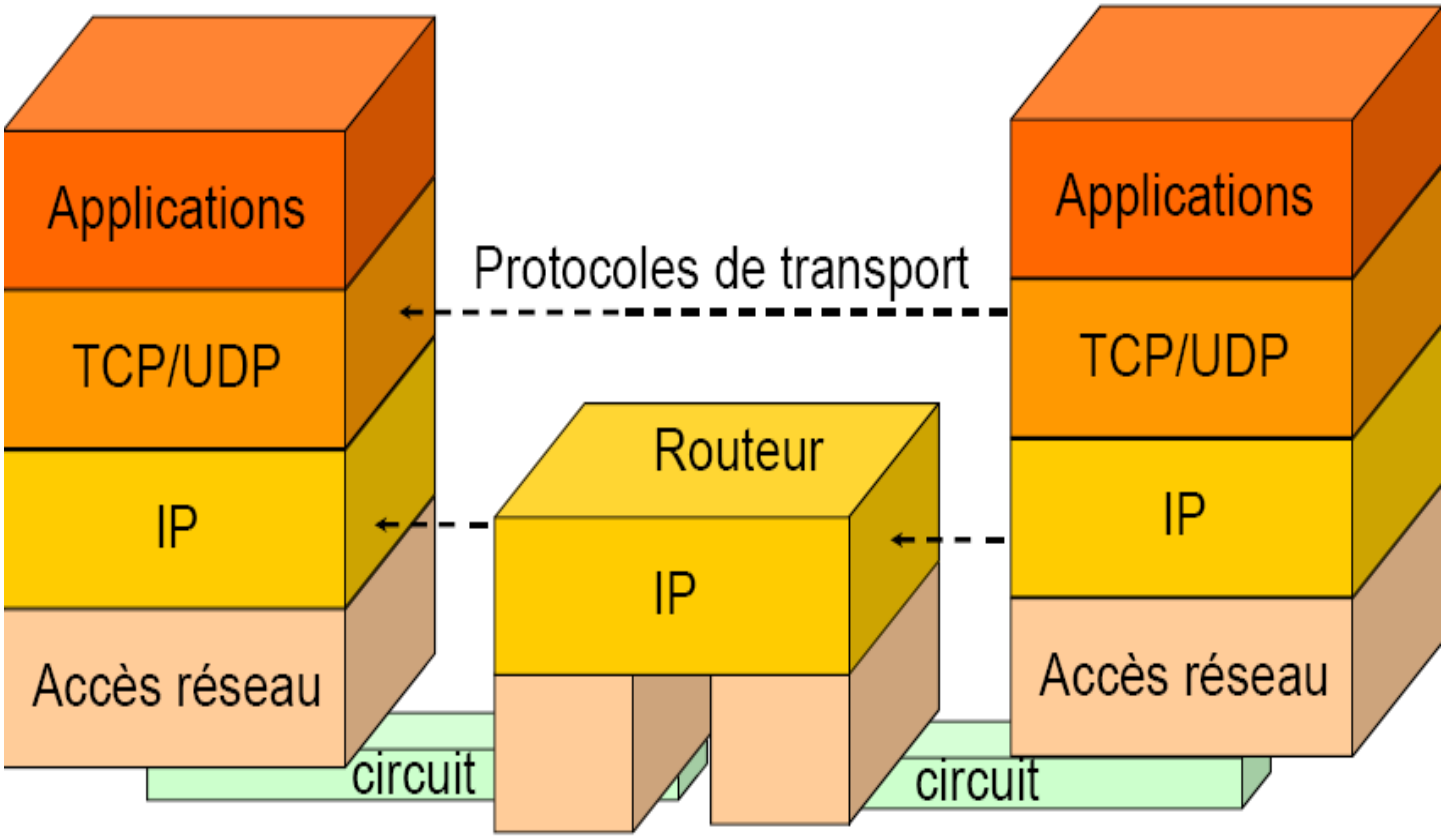
Transmission Control Protocol

Couche Transport

- Rappel



Couche Transport

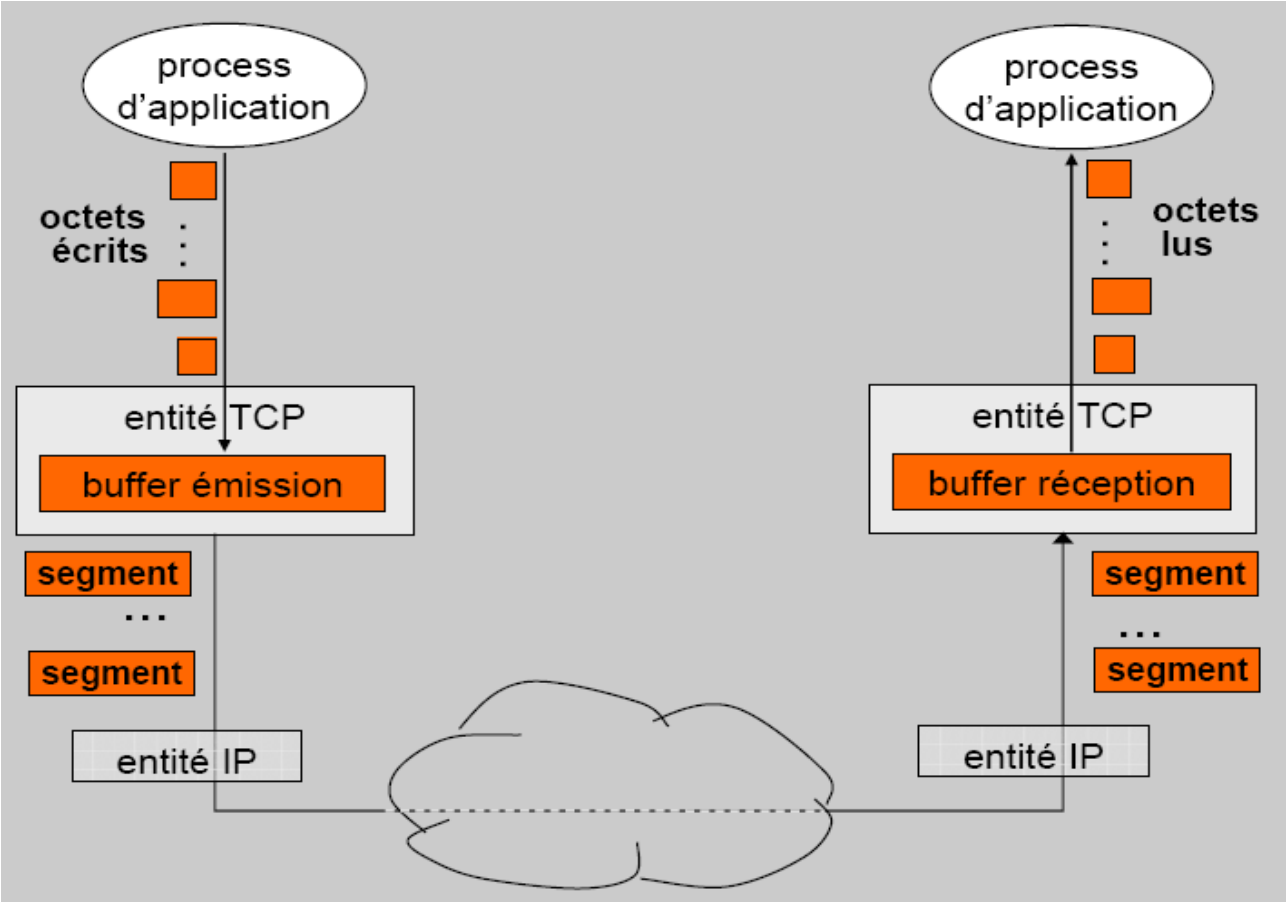
- Généralités
  - IP fait au mieux, « best effort »
  - Les protocoles de la couche transport subviennent aux défauts du protocole IP
  - UDP ou TCP ?
  - UDP ne corrige pas les défauts d'IP mais peu de services l'utilisent
  - TCP corrige les défauts d'IP
    - 90% du trafic applicatif utilise TCP
    - Utilisation optimale des ressources réseaux

Transmission Control Protocol

- Services offerts
  - Transfert fiable d'un flot d'octets entre processus applicatifs
    - TCP ne *tourne* pas dans les routeurs mais aux *extrémités* des réseaux (serveurs, imprimantes, etc) !
  - Point à point et full duplex en mode connecté
    - Ouverture de circuits virtuels à travers le réseau
  - Contrôle de flux et de congestion
    - Permettent à TCP et donc IP de s'adapter à tous types de réseaux sous jacents, rapides, lents, etc
  - S'adapte à différents types de transferts
    - Échanges interactifs
    - Transferts de masse
  - Pas de garantie de délai

Transmission Control Protocol

- Flot d'octets TCP

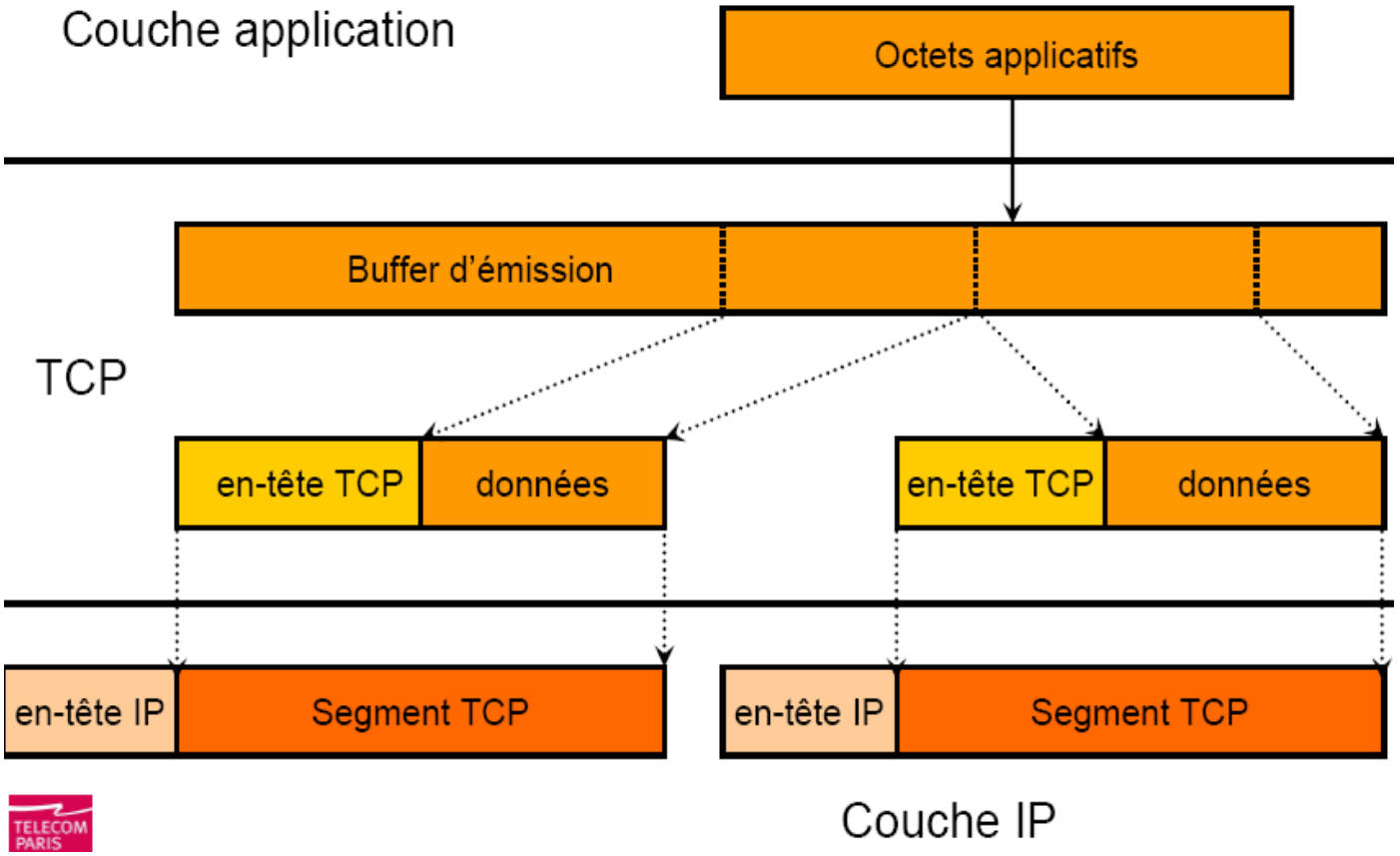


Transmission Control Protocol

- Principes et mécanismes
  - Protocole orienté connexion en trois phases
    - Ouverture de connexion
    - Transfert de données
    - Fermeture de connexion
  - Segmentation
    - Le flot d'octets est découpé en **segments** qui sont déposés dans des datagrammes IP transmis sur le réseau
    - Le champ protocole du datagramme IP = 6
  - Identification des segments par les numéros d'octets
  - Acquittement des segments reçus
    - => détection de perte et retransmission
  - Détection d'erreurs bit par calcul de redondance
  - Mesure intégrée du temps d'aller/retour (RTT)
  - Contrôle de flux par annonces de fenêtres
  - Contrôle de congestion sans signalisation réseau

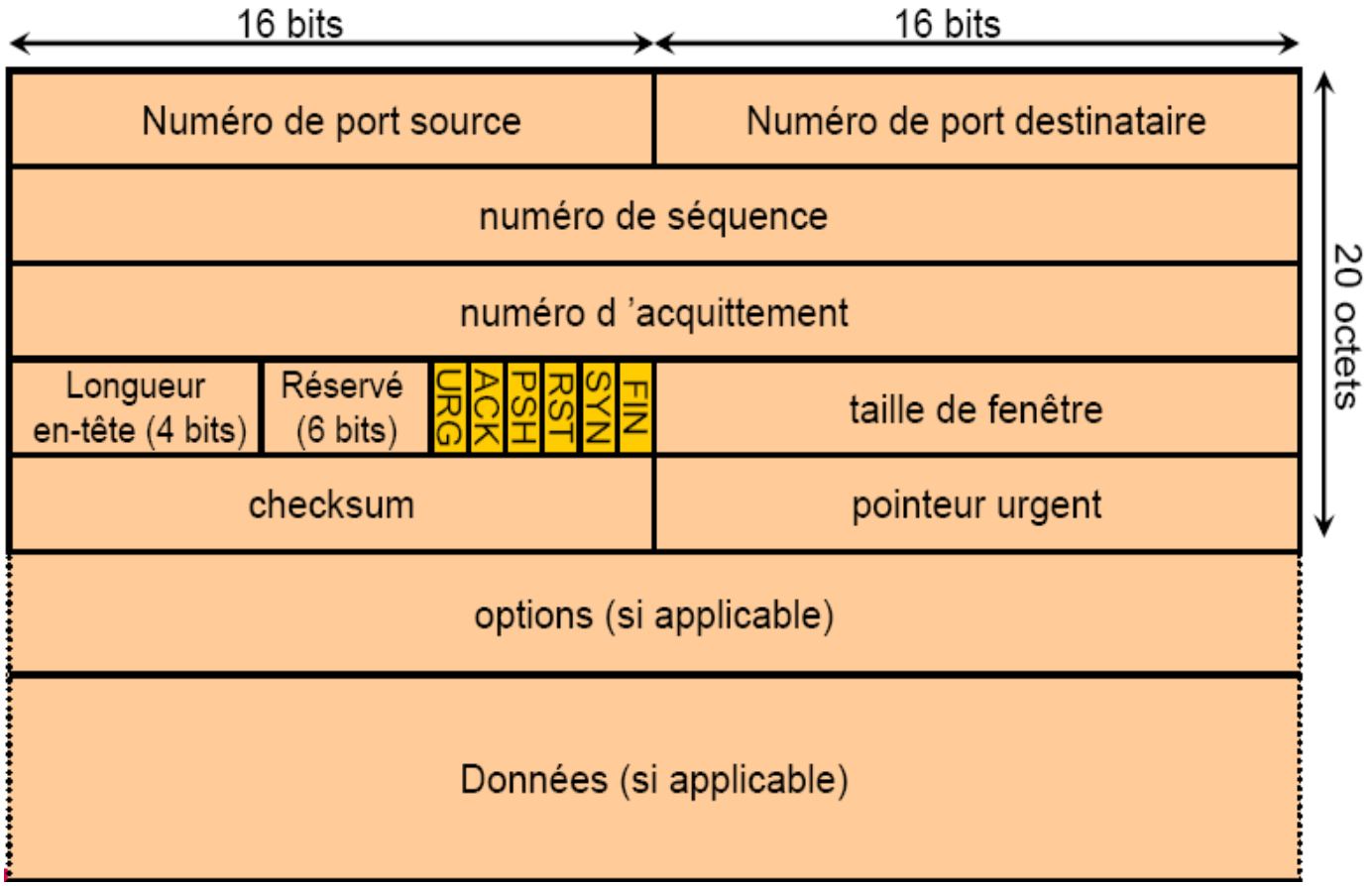
Transmission Control Protocol

- Segmentation



Transmission Control Protocol

- Entête d'un segment TCP



Transmission Control Protocol

- Entête d'un segment TCP
  - Numéro de port source
    - Identification de l'application source
  - Numéro de port destination
    - Identification de l'application destination
  - Couple Numéro de port + adresse IP
    - Socket sous Unix
  - Quadruplet ((IP1, PORT1),(IP2,PORT2))
    - Identification d'une connexion
  - Ports 1 à 1024
    - « Well known ports », principaux services serveurs
      - 20 et 21 : FTP, 22 : SSH, 23 : Telnet, 25 : SMTP,
      - 53 : DNS, 80 : HTTP, etc etc
  - Ports 1025 à 65535
    - Services serveurs secondaires mais connus, fixes et réservés
    - Alloués dynamiquement aux applications clientes

Transmission Control Protocol

- Entête d'un segment TCP
  - Numéro de séquence
    - Identification des octets envoyés par la source
      - Numéro du premier des octets des données contenues dans le segment
    - Pour un segment ayant SYN=1, c'est l'ISN
  - Numéro d'acquittement
    - Identification des octets reçus par le destinataire
      - Il s'agit du numéro du prochain octet que l'acquitteur attend, les octets précédents ayant tous été reçus
  - Numéro d'acquittement = Numéro de séquence du dernier segment reçu + taille du segment

Transmission Control Protocol

- Les drapeaux d'un segment TCP
  - Codés sur 1 bit chacun (0=inactif, 1=actif)
  - URG
    - Segment avec données urgentes
  - ACK
    - Le numéro d'acquittement est valide
  - PSH
    - Données à remettre (pousser) immédiatement à l'application
  - SYN
    - Segment d'ouverture de connexion portant un ISN (Initial Sequence Number)
  - RST
    - Fermeture immédiate de la connexion pour cause d'anomalie
  - FIN
    - L'émetteur n'envoie plus de données

Transmission Control Protocol

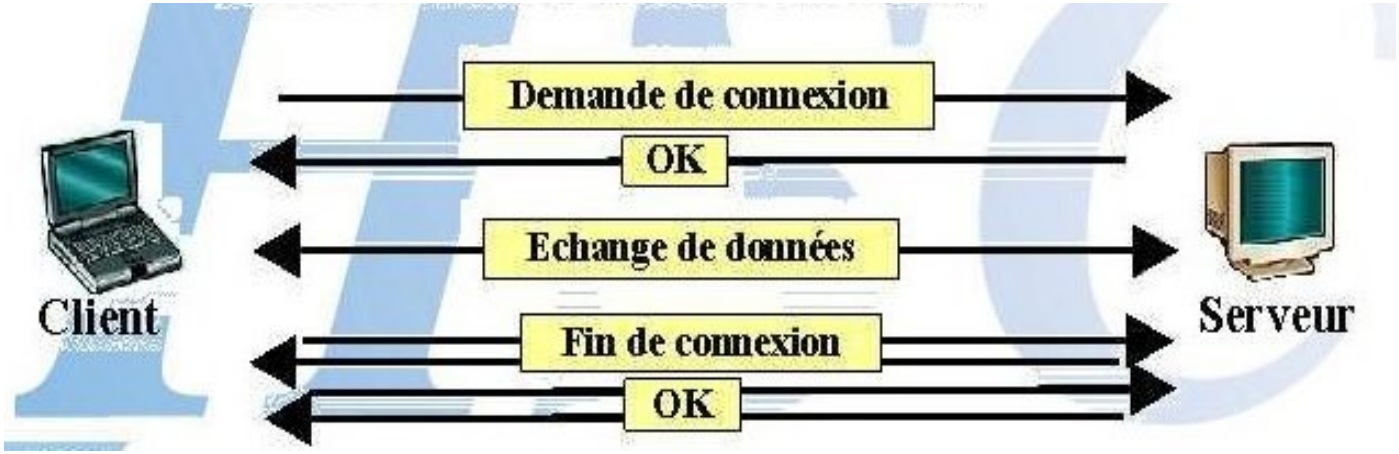
- Entête d’un segment TCP
  - Taille de la fenêtre
    - Utilisé pour le contrôle de flux
  - Checksum
    - Calculé par l’émetteur et vérifié par le récepteur
  - Pointeur urgent
    - Valide que si drapeau URG activé
  - Longueur entête
    - Pour déterminer la fin de l’entête et le début des données

Transmission Control Protocol

- Entête d'un segment TCP
  - Options
    - Parmi les options :
      - MSS : Maximum Segment Size : taille maximale en octets des données que l'entité TCP est prête à recevoir dans un segment, c'est donc la taille maximale en réception.
      - L'entité émettrice peut décider d'émettre des segments plus petits que le MSS pour s'adapter au MTU du réseau sous jacent et éviter la fragmentation IP
      - Exemple avec Ethernet :
        - » MTU Ethernet = 1500
        - » Entête IP = 20 octets
        - » Entête TCP = 20 octets
        - » ➔ MSS optimal = 1460 octets

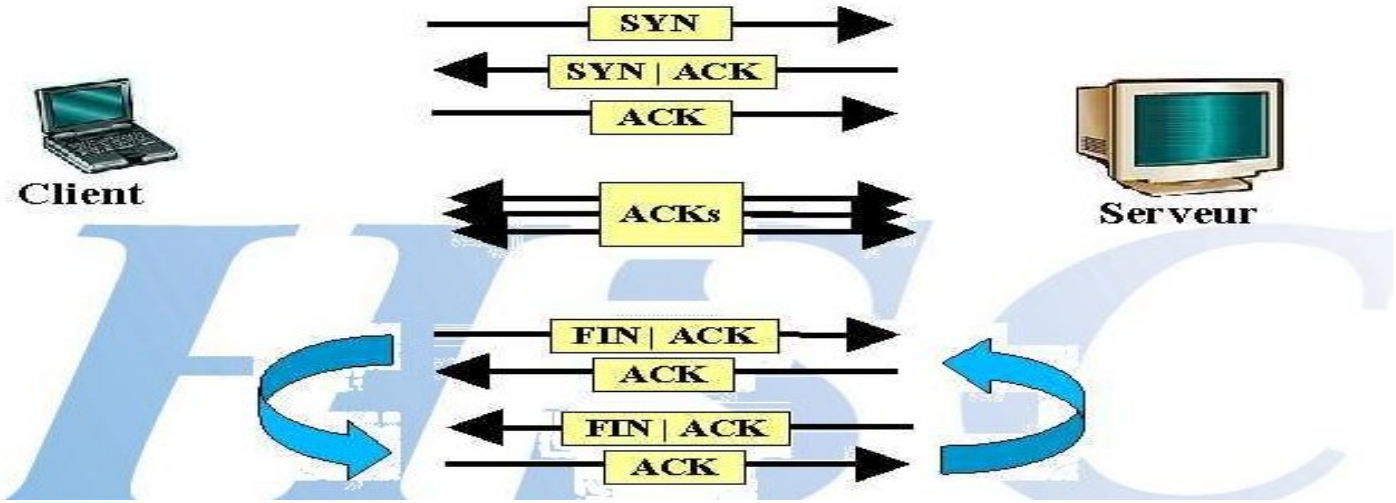
TCP : présentation d’une session

– Retransmission des paquets non acquittés
– n° de séquence unique pour chaque paquet TCP
– sens d ’établissement de la connexion
  • notion de drapeaux (SYN, ACK, FIN, RST, …)



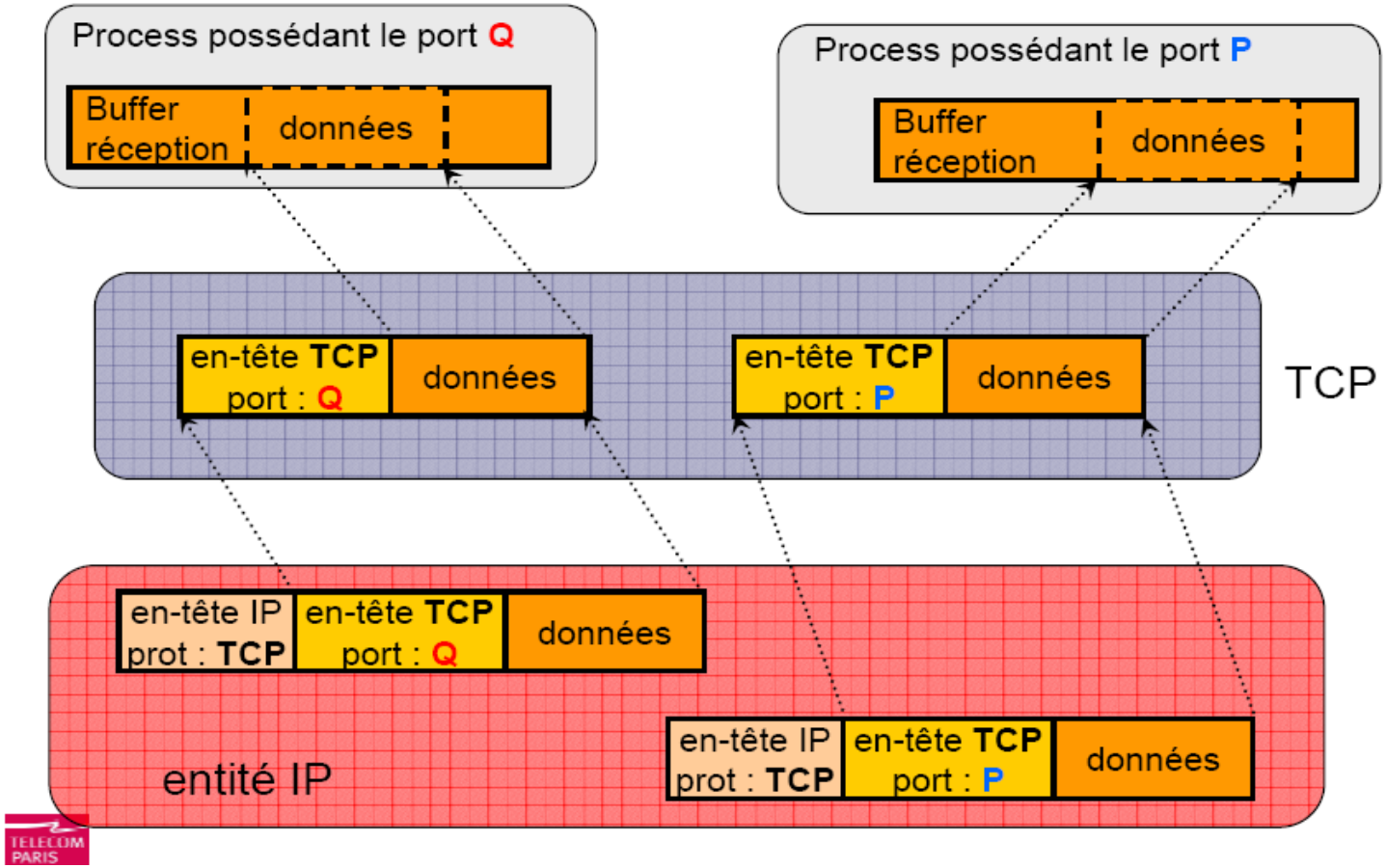
TCP : présentation d'une session

- Principaux drapeaux TCP (flags) :
  - SYN : établissement d'une connexion
  - ACK : acquittement d'un paquet
  - FIN : fin de session



Transmission Control Protocol

- Multiplexage et démultiplexage TCP



User Datagram Protocol

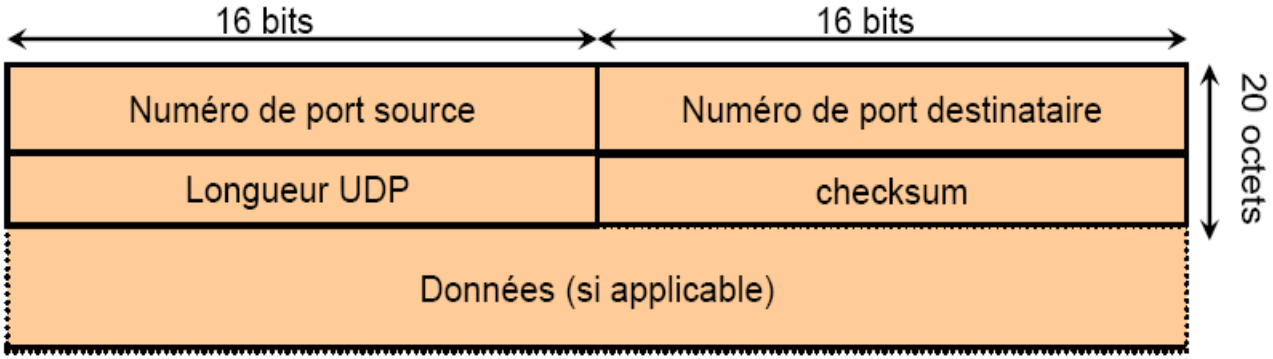
- Caractéristiques
  - Protocole de transmission de message datagramme
  - Pas de connexion
  - Pas de fiabilité
  - Pas de contrôle de flux ni de congestion
- A cause des contrôles et de sa fiabilité TCP peut engendrer de la latence pour certaines applications
  - UDP est alors utilisé pour laisser aux applications le nécessitant (par exemple temps réel) le soin de gérer les erreurs et les retransmissions

User Datagram Protocol

- Utilisations
  - Requêtes DNS
  - RPC (NFSv2)
  - Multicast
  - Temps réel audio/vidéo
- Protocole simple et utile
- Risque de congestion du réseau
  - Protocoles de multicast comme IGMP permettent de prévenir ces phénomènes

User Datagram Protocol

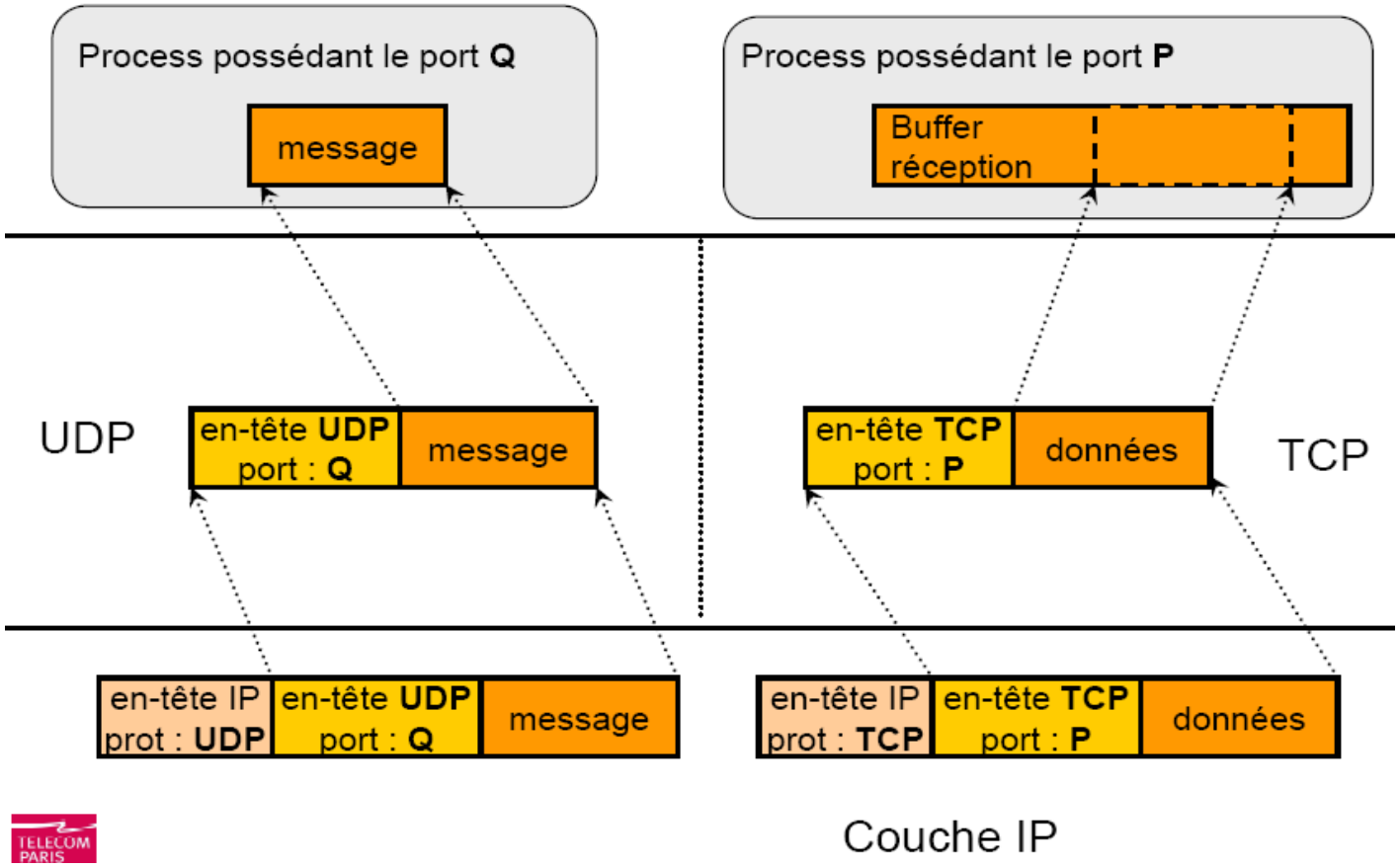
- Entête d'un segment UDP



- Numéros de port source et destinataire
  - identiques à TCP
- Longueur UDP
  - taille de l'en-tête + données
  - redondant avec les informations contenues au niveau IP
- Checksum : identique à TCP

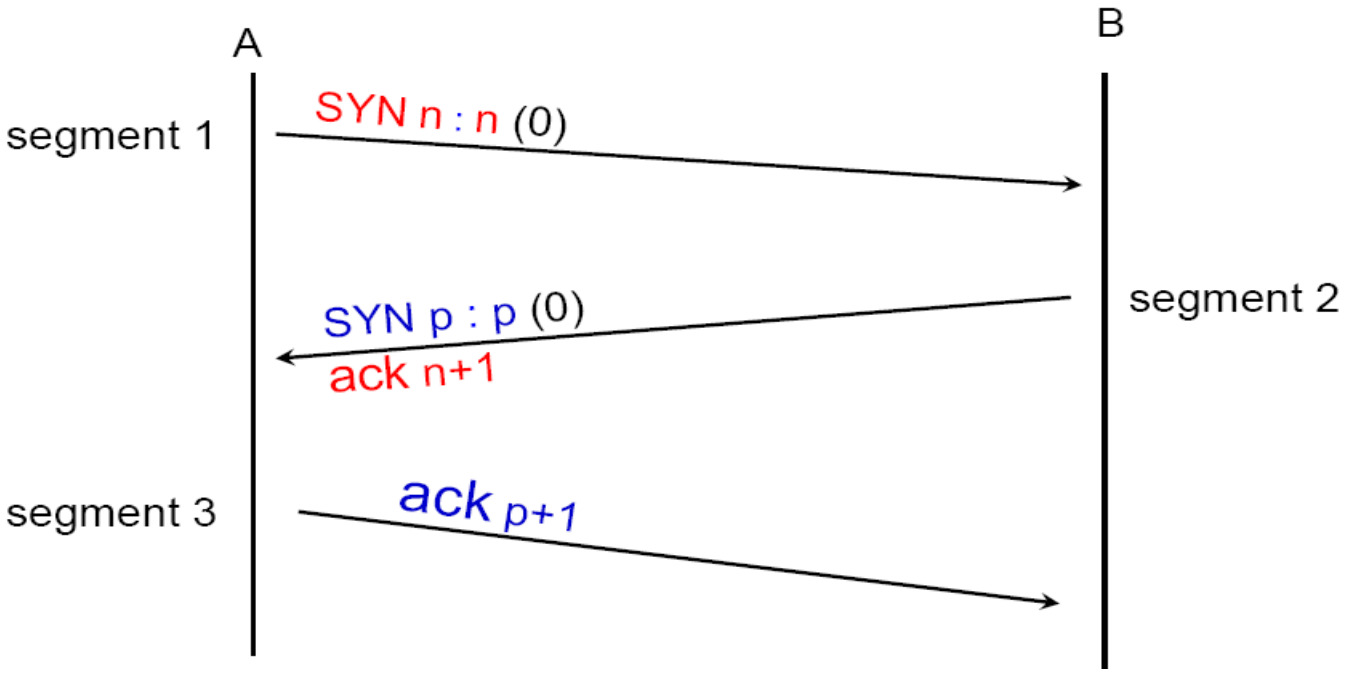
User Datagram Protocol

- Multiplexage et démultiplexage TCP et UDP



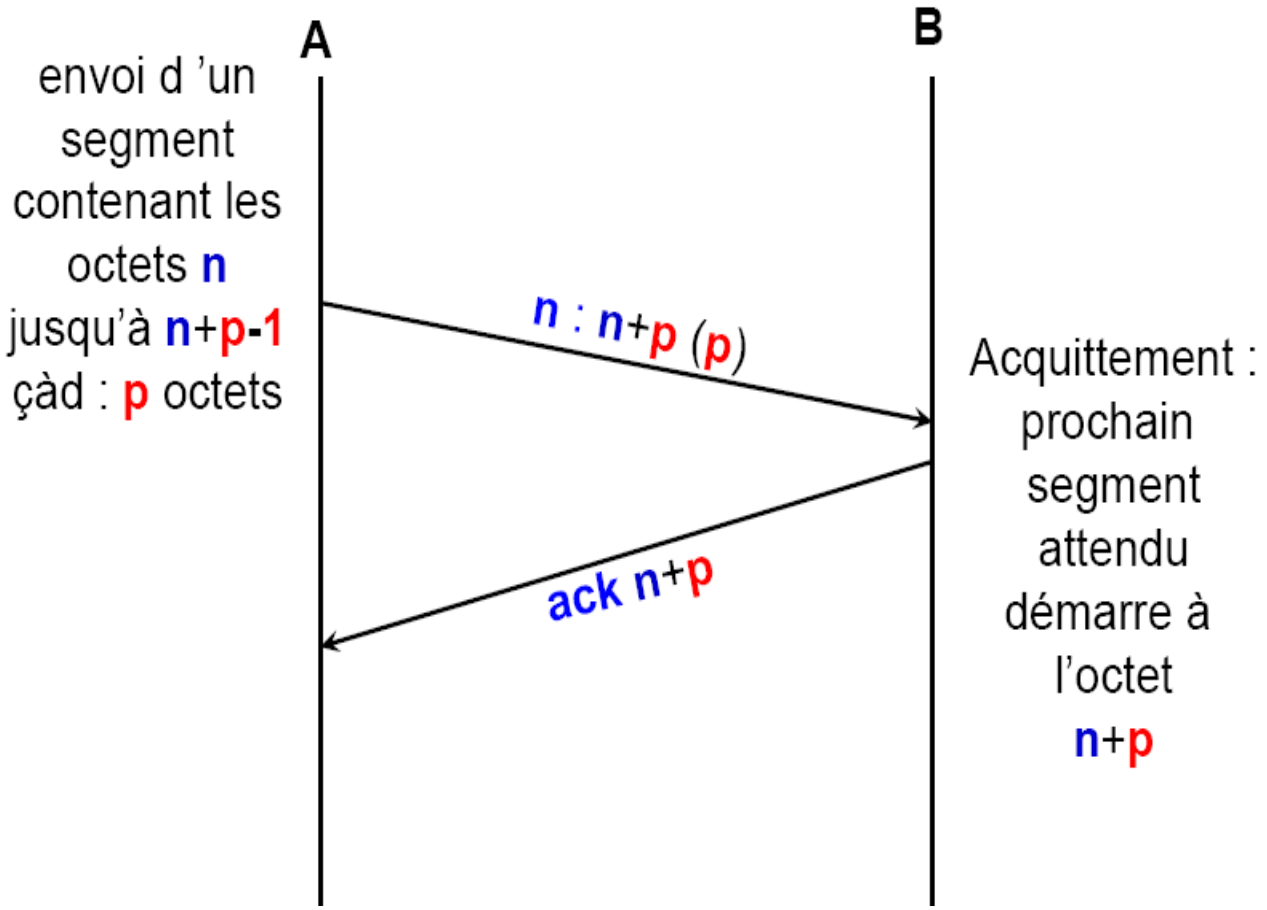
Transmission Control Protocol

- Ouverture de connexion TCP
  - Client A transmet à serveur B un segment de demande de connexion avec drapeau SYN activé et ISN à 0
  - Serveur B répond par un acquittement (ACK) du segment précédent et une ouverture de connexion (SYN)
  - Client A acquitte à son tour la confirmation du serveur B
  - La connexion bidirectionnelle est ouverte



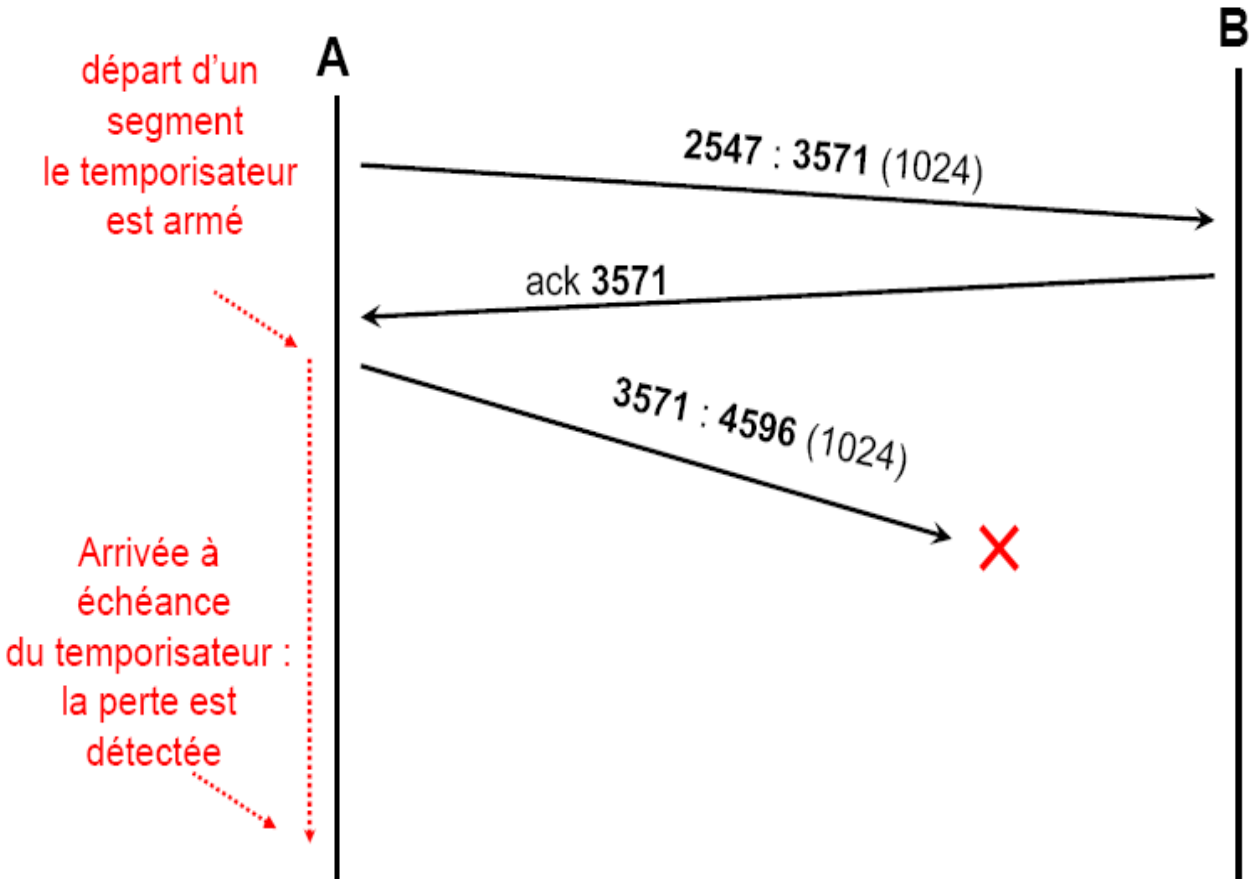
Transmission Control Protocol

- Transfert de données TCP



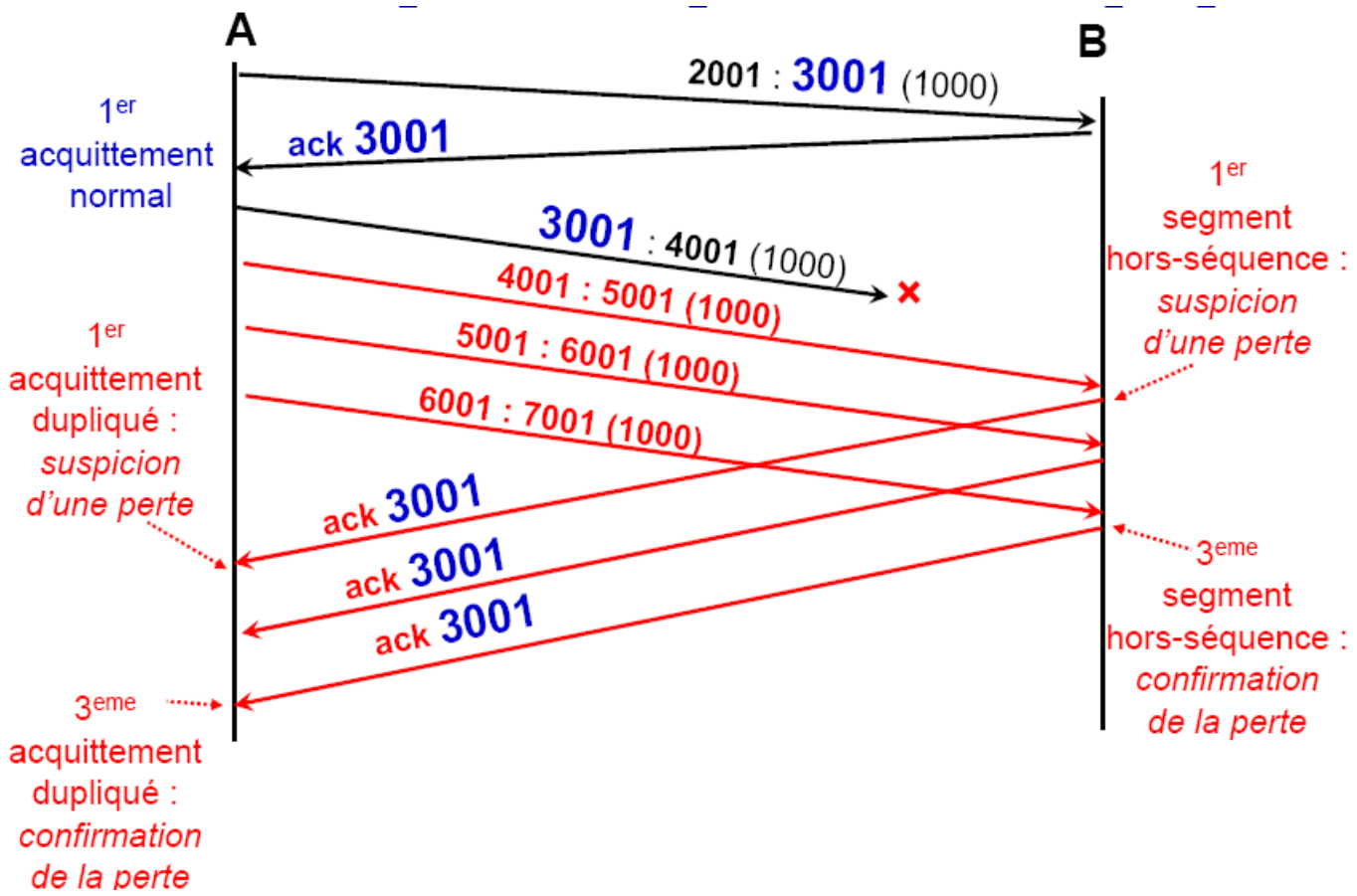
Transmission Control Protocol

- Détection de pertes TCP
  - Dépassement du délai d'attente



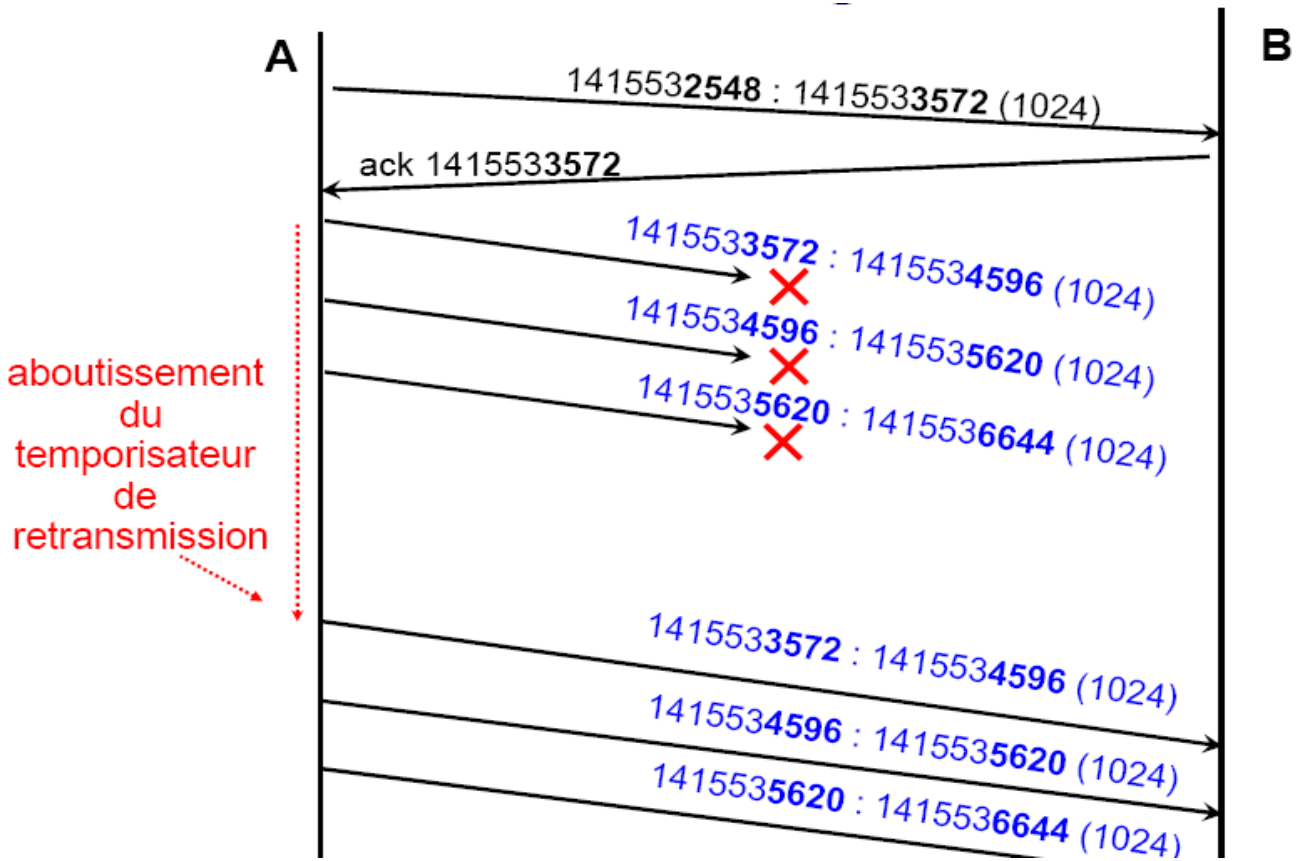
Transmission Control Protocol

- Détection de pertes TCP
  - Acquittements dupliqués



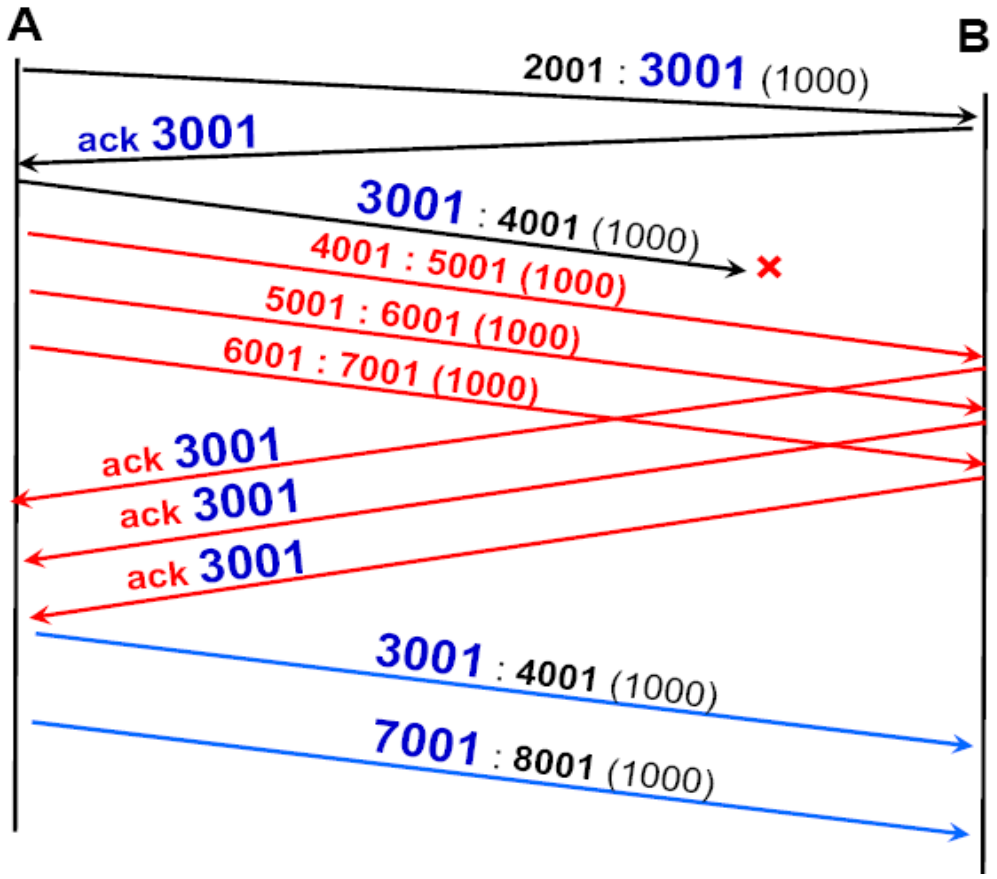
Transmission Control Protocol

- Retransmissions TCP
  - Retransmission complète après délai sans acquittement



Transmission Control Protocol

- Retransmissions TCP
  - Retransmission sélective après acquittement décalé

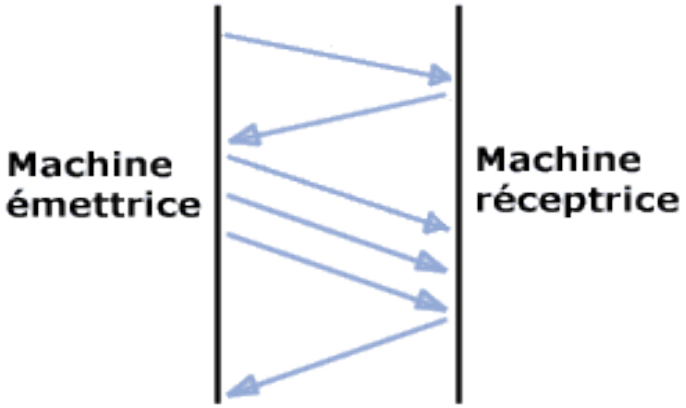


Transmission Control Protocol

- Contrôle de flux et de congestion TCP
  - Contrôle de flux
    - Évite que les destinataires ne soit engorgés par des sources trop rapides
  - Contrôle de congestion
    - Évite la transmission de segments sur le réseau alors qu'il est congestionné
    - Optimise le taux d'utilisation des ressources réseaux
  - Méthode des fenêtres glissantes
    - Mécanisme évolué permettant de contraindre l'émission des segments en jouant sur les quantités simultanées de données envoyées

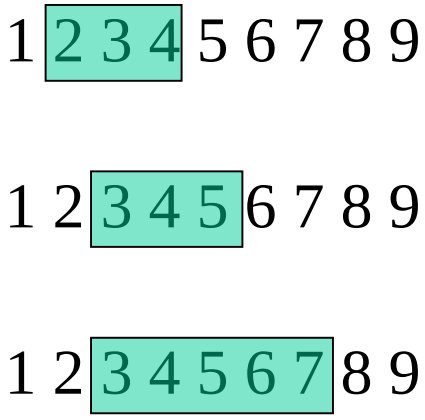
Transmission Control Protocol

- Fenêtre glissante TCP
  - Dans de nombreux cas, il est possible de limiter le nombre d'accusés de réception, afin de désengorger le réseau, en fixant un nombre de séquence au bout duquel un accusé de réception est nécessaire.
  - Ce nombre est en fait stocké dans le champ *fenêtre* de l'en-tête TCP/IP.
  - On appelle effectivement cette méthode "*méthode de la fenêtre glissante*" car on définit en quelque sorte une fourchette de séquences n'ayant pas besoin d'accusé de réception, et
  - La fenêtre se déplace au fur et à mesure que les accusés de réception sont reçus.



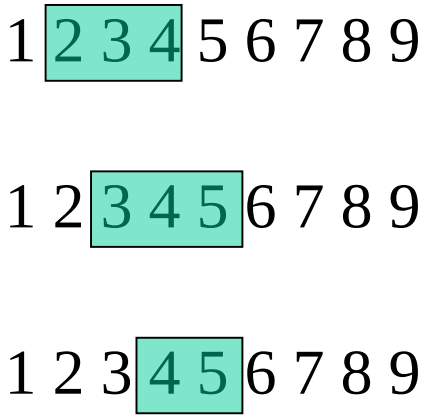
Transmission Control Protocol

- Fenêtre glissante TCP
  - De plus, la taille de cette fenêtre n'est pas fixe.
  - En effet, le serveur peut inclure dans ses accusés de réception en stockant dans le champ fenêtre la taille de la fenêtre qui lui semble la plus adaptée.
  - Ainsi, lorsque l'accusé de réception indique une demande d'augmentation de la fenêtre, le client va déplacer le bord droit de la fenêtre.
  - Le nombre de segments transmis augmente donc.



Transmission Control Protocol

- Fenêtre glissante TCP
  - Par contre, dans le cas d'une demande de diminution de la fenêtre, le client ne va pas déplacer le bord droit de la fenêtre vers la gauche mais attendre que le bord gauche avance (avec l'arrivée des accusés de réception).
  - Le nombre de segments transmis diminue donc.



Transmission Control Protocol

- Avantages de la fenêtre glissante TCP
  - La congestion est détectée automatiquement par la source en fonction de l'état des fenêtres.
  - Il y a toujours un démarrage lent (slow start) calculé sur la taille du MSS
  - Augmentation exponentielle de la fenêtre
  - Ralentissement de l'augmentation pour éviter les engorgements
  - Possibilité de recouvrement rapide de la taille antérieure de la fenêtre après une congestion
  - Par ailleurs un temps d'aller/retour (RTT) est recalculé toutes les 500 ms sur la base du délai observé entre en envoi de segment et son acquittement.

Transmission Control Protocol

- Fin de connexion TCP
  - Confirmation de fin de connexion : client ET serveur