TCP et les nouvelles protocoles de transport

Source: Cours de Jean-Patrick Gelas, Lyon, France

Introduction

- Le protocole TCP standard (*TCP Reno*) est un protocole de transport fiable qui fonctionne bien dans les réseaux traditionnels.
- Cependant, les expériences et l'analyse de ce protocole montre qu'il n'est pas adapté à toutes les **applications** et à tous les **environnements,** par exemples :
  - les communications interactives ;
  - le transfert haut débit de grosse quantité de données (*bulk data transfer*) ;
  - les réseaux dont produit débit et temps d'aller/retour (throughput x *RTT (round-trip time)*) sont grand.
  - réseaux sans-fils.

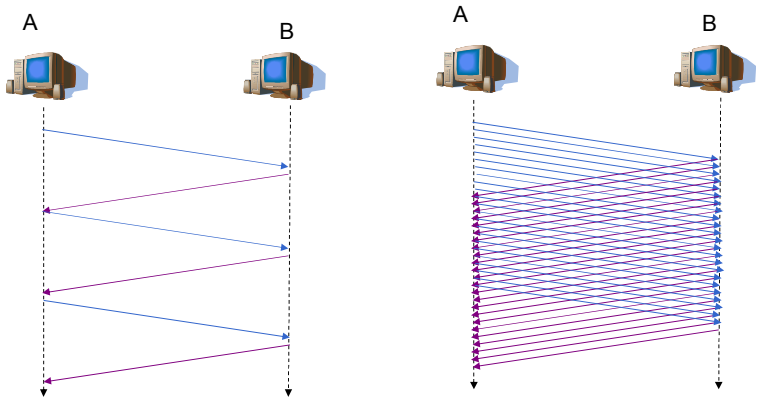
Deux composants

- Contrôle de flux : comment faire pour être sur que le récepteur reçoit aussi vite que l'on émet ?
- Contrôle de congestion : comment être sur que le réseau délivre les paquets au récepteur ?

Vocabulaire utilisé dans
les protocoles de contrôle de congestion

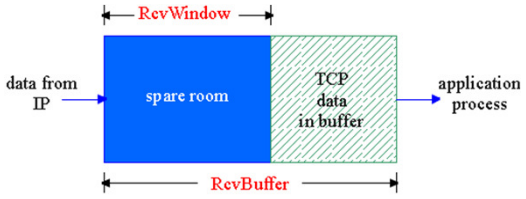
- cwnd – congestion window
- RTT – Round trip Time
- RTO – Retransmission Timeout
- MSS – Maximum Segment Size
- DUPACK – ACK dupliqué
- ssthresh – Slow Start Threshold

Rappel sur le contrôle de flux de TCP



Rappel sur le contrôle de flux de TCP

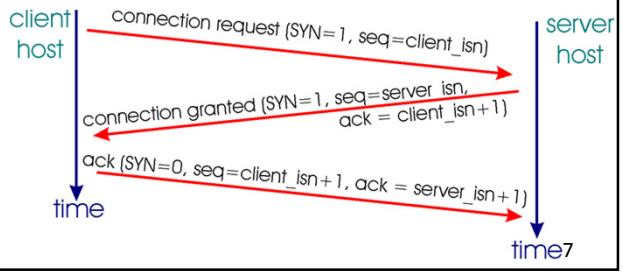
- Chaque pôle d'une connexion à un tampon de réception.
- Sans service de contrôle de flux un émetteur pourrait rapidement saturer le tampon de réception (*RcvBuffer*) dont la taille est fixe.
- L'expéditeur est informé de la taille variable de la **fenêtre de réception** (*RcvWindow*) du destinataire.
- Le récepteur informe l'expéditeur de l'espace disponible en insérant la variable *RcvWindow* dans la fenêtre de réception de tous les segments qu'il lui envoie.



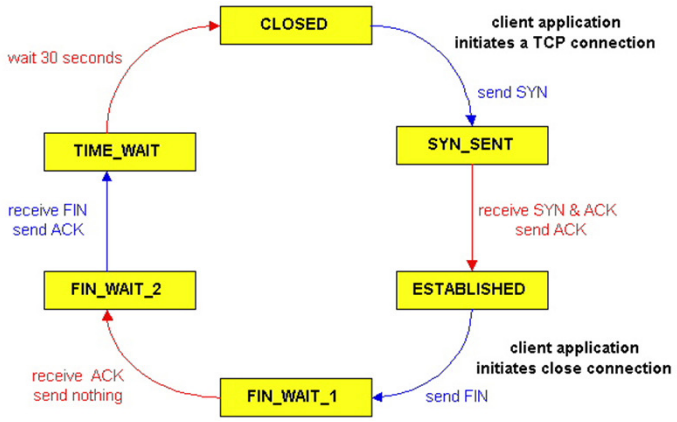
Rwnd = RcvBuffer-[LastByteRcvd - LastByteRead]

Rappel sur la gestion de connexion TCP
Établissement de connexion en trois étapes

- La procédure d'établissement d'une connexion TCP constitue une partie non négligeable du temps de réponse perçu par les utilisateurs.
- Etape 1 : **segment SYN** (*bit SYN=1*), le client choisi un numéro de séquence initial (seq = *client_isn*).
- Etape 2 : **segment SYNACK**. Le récepteur alloue un tampon et des variables, renvoie un datagramme avec *SYN=1*, seq = *server_isn*, ack = *client_isn+1*
- Etape 3 : Le client alloue un tampon des variables, et accuse réception de cette autorisation de connection.
  SYN=0, seq = *server_isn+1,*
  ack = client_isn+1

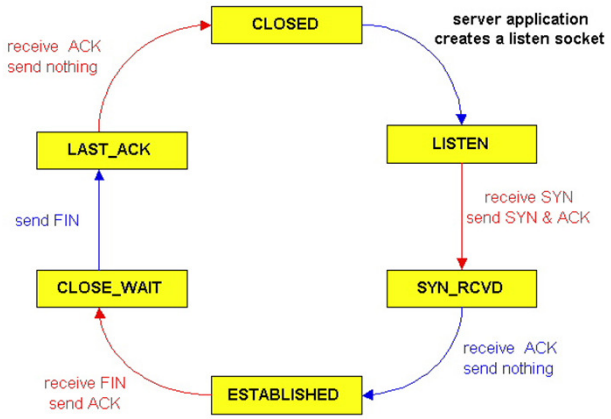


Rappel sur la gestion de connexion TCP : Etats TCP



*Etats TCP traversé par le pôle **client** TCP.*

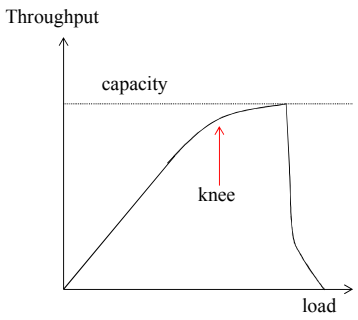
Rappel sur la gestion de connexion TCP : Etats TCP



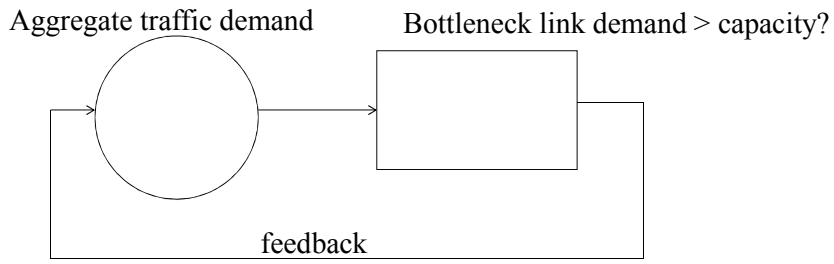
*Etats TCP traversé par le pôle **serveur** TCP.*

Le contrôle de congestion de TCP

- L'objectif principal du contrôle de congestion est d'éliminer la congestion.
- Lorsque la demande totale du trafic excède la capacité (throughput) du lien, cette demande doit être réduite.
- Sinon, *congestion collapse* potentiel... (événement couramment observé dans les années 80 avant l'ajout du contrôle de congestion)



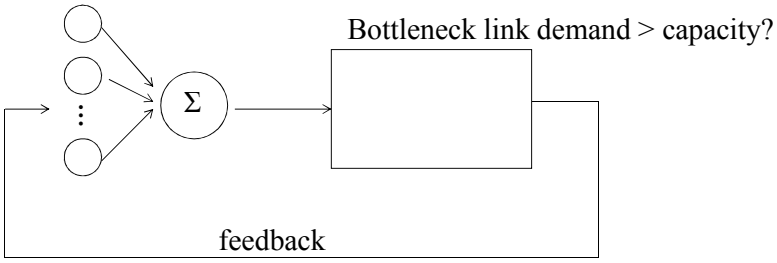
Modèle de base du contrôle de congestion



- Quel est le type de *feedback* ?
- Est-ce stable ?

Contrôleurs multiple

- Dans le cas du contrôle de congestion de réseaux (plusieurs connections partageant le même lien), il y a plusieurs contrôleurs



- Est-ce équitable ? (fairness)

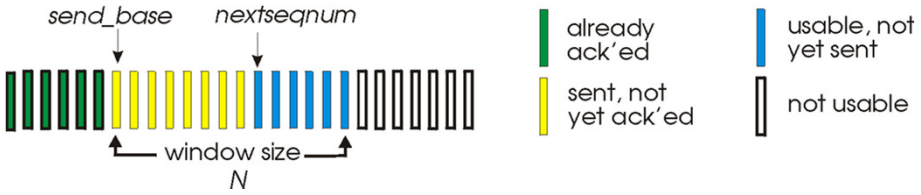
Le contrôle de congestion de TCP

- Puisque IP ne fournit aucune information explicite aux terminaux, TCP a nécessairement recours à une approche de **bout-en-bout** (plutot qu'à un contrôle de congestion assisté par le réseau).
- La vitesse d'émission est régulée en fonction du niveau de congestion **perçu**.
- Régulation du taux d'envoi :
  - La variable **fenêtre de congestion** (*CongWin* ou *cwnd*) impose une limite au rythme auquel l'expéditeur est autorisé à charger des données sur le réseau.
  - Le volume de donnée non-confirmées ne peut dépasser le minimum de *CongWin* et *RecvWin*.

  $LastByteSent - LastByteAcked = \min \{CongWin, RecvWin\}$
- Le "phénomène de perte" est identifé/perçu soit par :
  - l'expiration du temps imparti (*timeout*), soit par
  - la réception de trois ACK identiques de la part du destinataire,

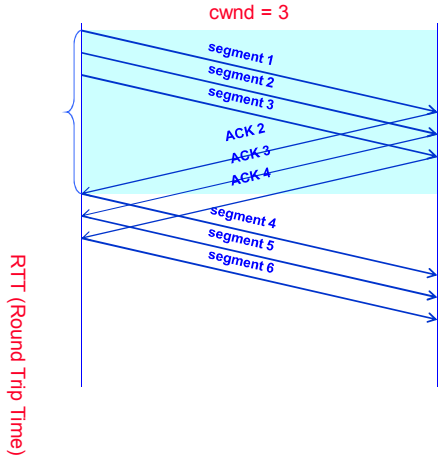
et interprété comme un indice de congestion sur le parcours vers le destinataire.

Rappel sur le contrôle de congestion de TCP (suite)



Taille de fenêtre et débit

- De la taille de fenêtre de congestion dépend le débit !
- Throughput = cwnd / RTT



Rappel sur le contrôle de congestion de TCP :

Départ lent (*slow start*)

- Au début d'une connexion, *CongWin* = 1 MSS, donc
  un taux d'envoi ~= MSS/RTT (ex: 500 bytes / 200 ms = 20 kbit/s).
- Au début, l'émetteur à en fait recours à une accélération exponentielle. Pour chaque ACK reçu **cwnd = cwnd + 1**.
- Autrement dit, cela consiste à doubler la taille de la *CongWin* à chaque RTT.

  1 MSS, 2 MSS, 4 MSS,...
- Cette technique est utilisé jusqu'à :
  - ce qu'il y ait une perte, moment auquel *CongWin* est **divisé par 2** ou
  - qu'on atteigne une valeur seuil ***slowstart treshold***.

*et passe en mode de **progression linéaire** ou évitement de congestion (congestion avoidance).*

Rappel sur le contrôle de congestion de TCP :

Accroissement additif et décroissance multiplicative

- L'idée de base est de faire en sorte que l'expéditeur réduise son taux d'envoi en diminuant la taille (par 2) de sa fenêtre de congestion dés qu'un phénomène de perte se déclare (ex: 20ko -> 10ko -> 5 ko ... -> 1 MSS (*Maximum Segment Size*)) : **cwnd = cwnd / 2**

- Si le réseau ne présente pas de congestion, une “certaine” valeur de débit doit être disponible pour la connection TCP.
- TCP procède a un agrandissement progressif de sa *CongWin*, “sondant” scrupuleusement toute éventuelle fraction de débit disponible sur le chemin de bout-en-bout : Pour chaque
  ACK reçu **cwnd = cwnd + 1/cwnd**.
- autrement dit, TCP ajoute environ 1 MSS à chaque temps de trajet aller/retour (RTT) tant qu'aucun phénomène de perte ne se déclare.

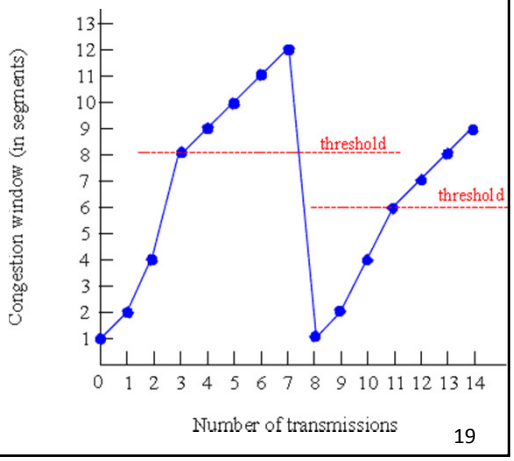
Rappel sur le contrôle de congestion de TCP :

Accroissement additif et décroissance multiplicative

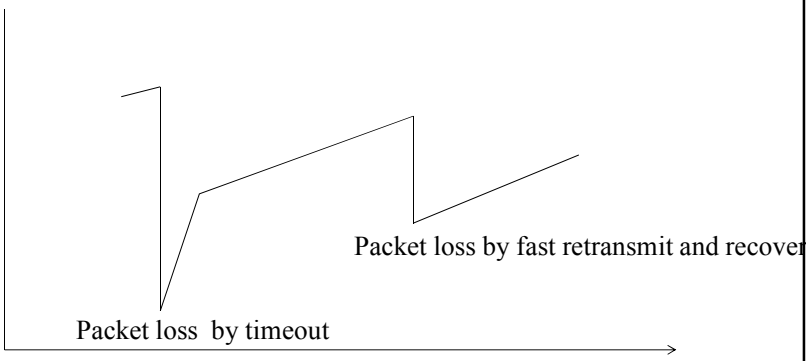
- L'algorithme de congestion de TCP est donc appelé AIMD (*Addititve Increase, Multiplicative Decrease*).
- La phase de croissance linéaire -> phase d'évitement de congestion (*congestion avoidance*).

Réaction au phénomène d'expiration du temps imparti ou du triple ACK

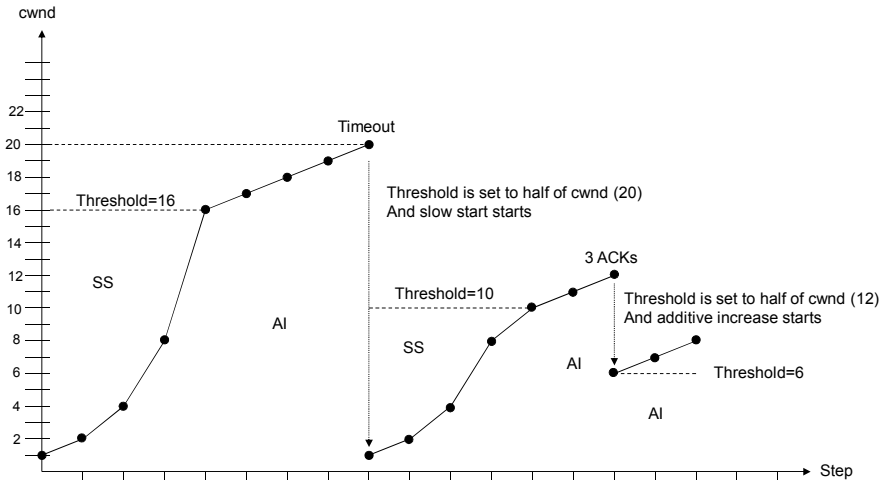
- Le phénomène de congestion de TCP ne réagit pas de la même manière à un phénomène de perte détecté par :
  - l'expiration du temps imparti (*timeout*), ou
  - l'arrivée de 3 ACK identiques
- Pour le cas ***Timeout*** : CongWin=1, agrandissement exponentielle jusqu'à une valeur seuil (*SSTreshold*), puis agrandissement linéaire.
- Pour le cas **Triple ACK** : L'annulation de la phase de départ lent après un triple ACK est appelé récuperation rapide (*fast recovery*).
- *SSThreshold* initial par défaut = 64 ko, en cas de perte *SSTreshold* = *CongWin* / 2



Comportement dynamique

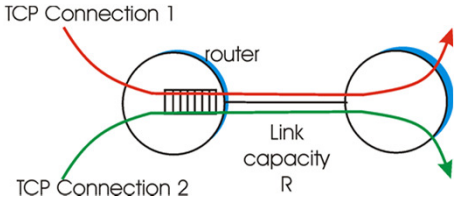


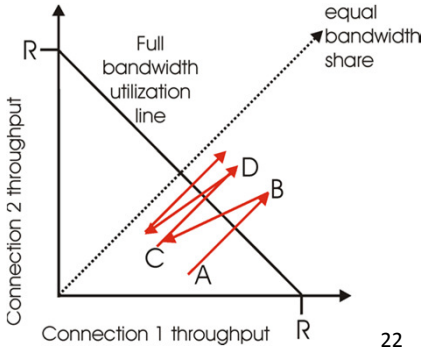
Comportement dynamique



Equité (*fairness*)

- Soit *K* connections TCP empruntant toutes une même liaison "faible" (goulet d'étranglement) d'un débit de *R* bit/s.
- On suppose qu'il n'y a pas de flux UDP.
- Un système de contrôle de congestion est dit équitable si le débit moyen de chaque connexion ~= *R* / *K*
- AIMD est-il équitable ? (connexions pas forcément établi en même temps, fenêtre de taille différente).





État de l'art
(non exhaustif !)

- **Variantes de TCP :**

  HighSpeed TCP, Scalable TCP, Fast TCP, BIC and CUBIC, TCP Weswood+,...

- **Protocoles basés sur UDP :**

  UDP Lite, Reliable Blast UDP, Tsunami, UDT,...

- **Protocoles exigeant l'aide des routeurs :**

  XCP, CADPC/PTP,...

- **Autres :**

  SCTP, DCCP,...

Conclusion

- Face à la mise en place de plus en plus importante de liens longue distance à très haut débit (débit x RTT important), la mise au point de nouveaux protocoles capable d'exploiter ces liens est indispensable.
- Difficile de créer un protocole capable de répondre aux exigences de toutes les applications existante et à venir...
- Beaucoup d'activités de recherche dans ce domaine suivi de trés prêt par tous les équipementiers réseaux.