Mécanismes de fiabilité

Introduction aux réseaux locaux

Plan de la présentation

- Mécanismes pour la fiabilité du transport

Mécanismes pour la fiabilité du transport

Rappel

- La couche transport se situe entre la couche application et la couche réseau et s'occupe du transport des données entre deux processus (applications)
- Les données envoyés par cette couche sont des segments qui englobent les messages de l'application.
- La couche transport pourvoit un lien logique entre deux applications.
- Il existe deux protocoles distincts pour la couche transport d'Internet:
  - UDP (User Datagram Protocol)
  - TCP (Transmission Control Protocol)

TCP vs UDP

« I know this great UDP joke but you might not get it. »

« I'm going to keep telling you this TCP joke until you get it. »

Les différents mécanismes

- Pour assurez le transfert de données de la source vers la destination, plusieurs mécaniques de contrôles sont utilisés:
  - Dans le cas où les données peuvent être corrompues
  - Somme de contrôle
  - Accusé réception
  - Séquence
  - Dans le cas où les paquets sont perdus
  - Chronométrage
  - Fenêtre et tuyau

Somme de contrôle

- Pour s’assurer rapidement que le paquet contient la bonne information il faut valider qu’il n’y est:
  - Aucune corruption
  - Aucune perte ou ajout de données
- Pour se faire, une somme de contrôle est introduite dans l’en-tête du segment.
- Cette somme est valide pour UDP et TCP
  - UDP peut décider d’ignorer un paquet erroné ou envoyer un message à l’application

Comment calculer la somme ?

- Pour calculer la somme de contrôle d’un segment:
  - Il faut additionner en binaire, sur 16 bits, l’ensemble du segment et d’utiliser son complément à la fin.
  - Si, lors de l’addition, il y a une dernière retenue (débordement du dernier bit), la rapporter au début.
  - Si le message est sur un nombre impair d’octets, ajouter un zéro.
  - Lors du calcul de la somme pour l’en-tête du segment on utilise la somme ‘temporaire’ de zéro.

Exemple de somme

Message
0101
0010
0001
1010
1000

Somme sur 4 bits
0101 + 0010
0111 (Somme des deux premiers)
0111 + 0001 (on continue avec le troisième)
1000 (somme des trois premiers)
1000 + 1010 (Somme avec le quatrième)
0011 (ATTENTION, retenu à reporter)
0011 + 1000 (Somme avec le dernier)
1011 (Il faut faire le complément)
0100 (Somme de contrôle, rapportée dans l'en-tête)

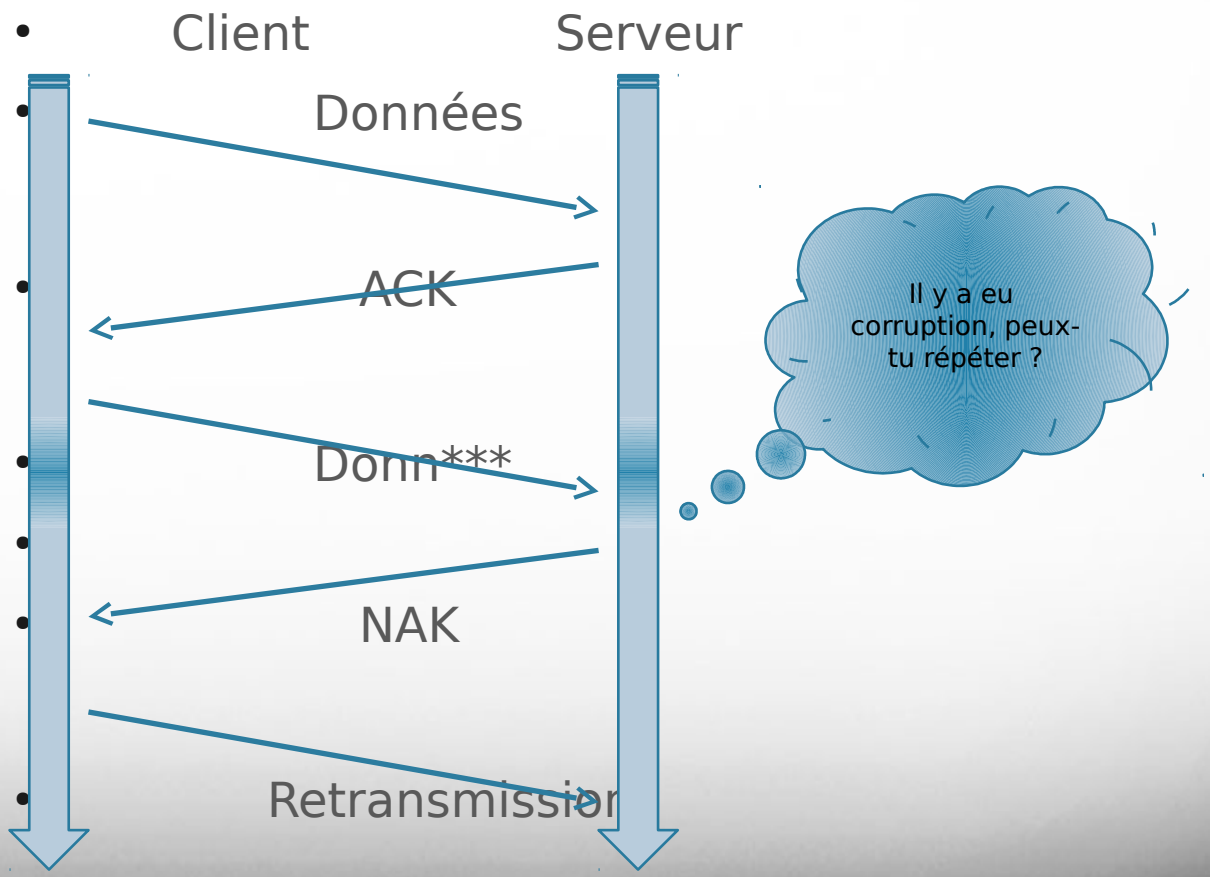
Validation des données

- Lors de la réception du paquet, la somme est recalculée est validée.
- Si la somme ne correspond pas, on assume une perte ou une altération des données.
- Pour permettre à l’expéditeur de savoir que la réception fut bien faite, un accusé réception sera envoyé.
  - Lui aussi avec une somme de contrôle…

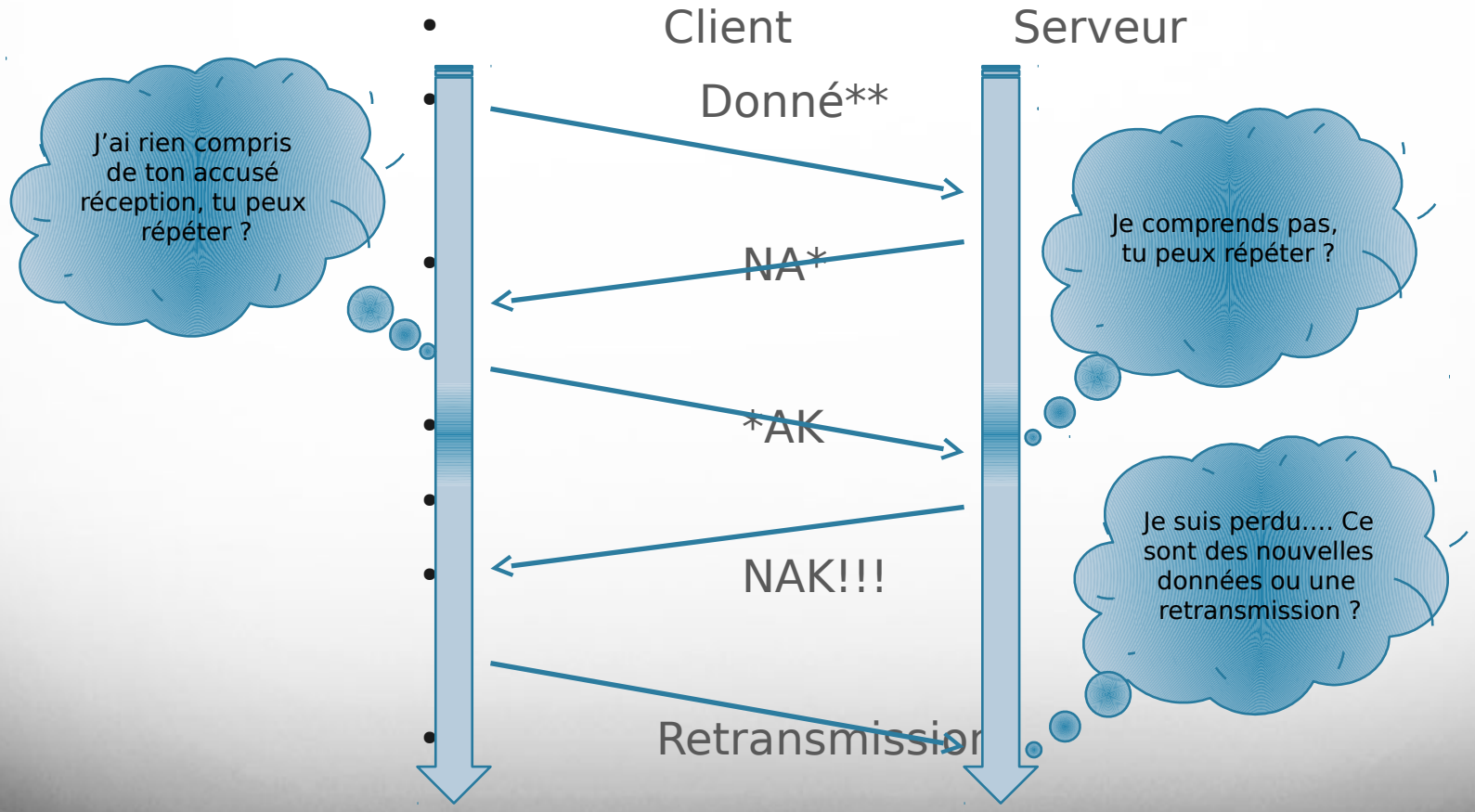
Accusés réception

- Dans une conversation normale, nous avons deux types d'accusés réception pour expliquer si le message s'est bien rendu:
  - Le 'Ok' (Oui, j'ai compris, continue)
  - Le message c'est bien rendu
  - Le 'Peux-tu répéter?'
  - Il y a eu corruption du message. Nous désirons recevoir le message à nouveau.
- Dans le cas de message réseau, ils seront étiquetés ACK (pour acknowledgement) et NAK (pour Negatively acknowledge)

Exemple d’accusé simple



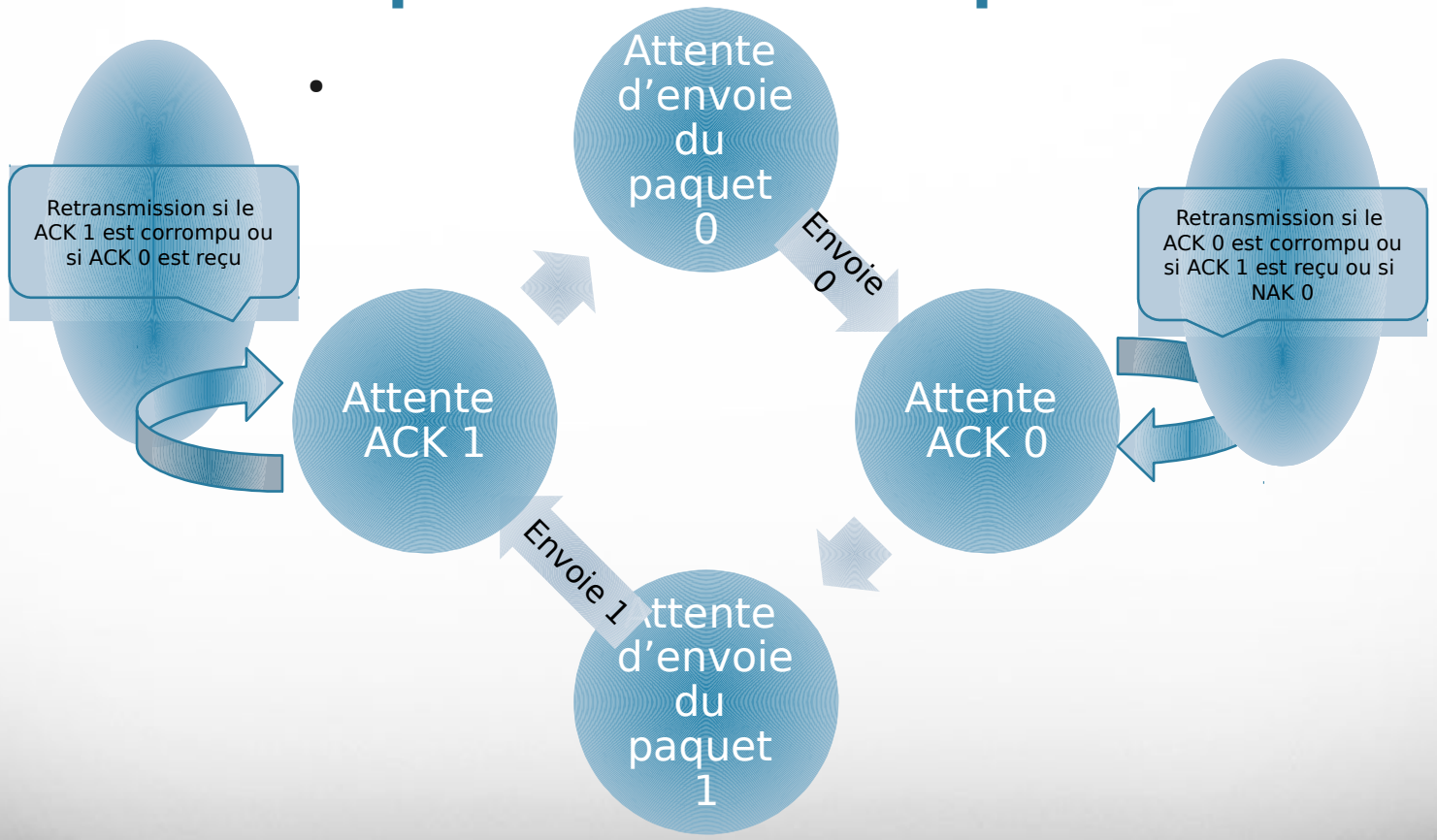
Authentification des accusés?



La séquence

- Si l’envoyeur ne comprend pas l’accusé réception, il doit retourner le message pour éviter de tomber dans une boucle sans fin de ‘NAK’
- Par contre, étant donné que le récepteur ne sait pas si son ACK ou NAK fut bien reçu, il ne peut pas savoir si le nouveau message est une retransmission ou un nouveau paquet.
- Pour différencier ces messages, l’expéditeur spécifiera un ordre, une séquence à ses paquets.

Exemple de séquence



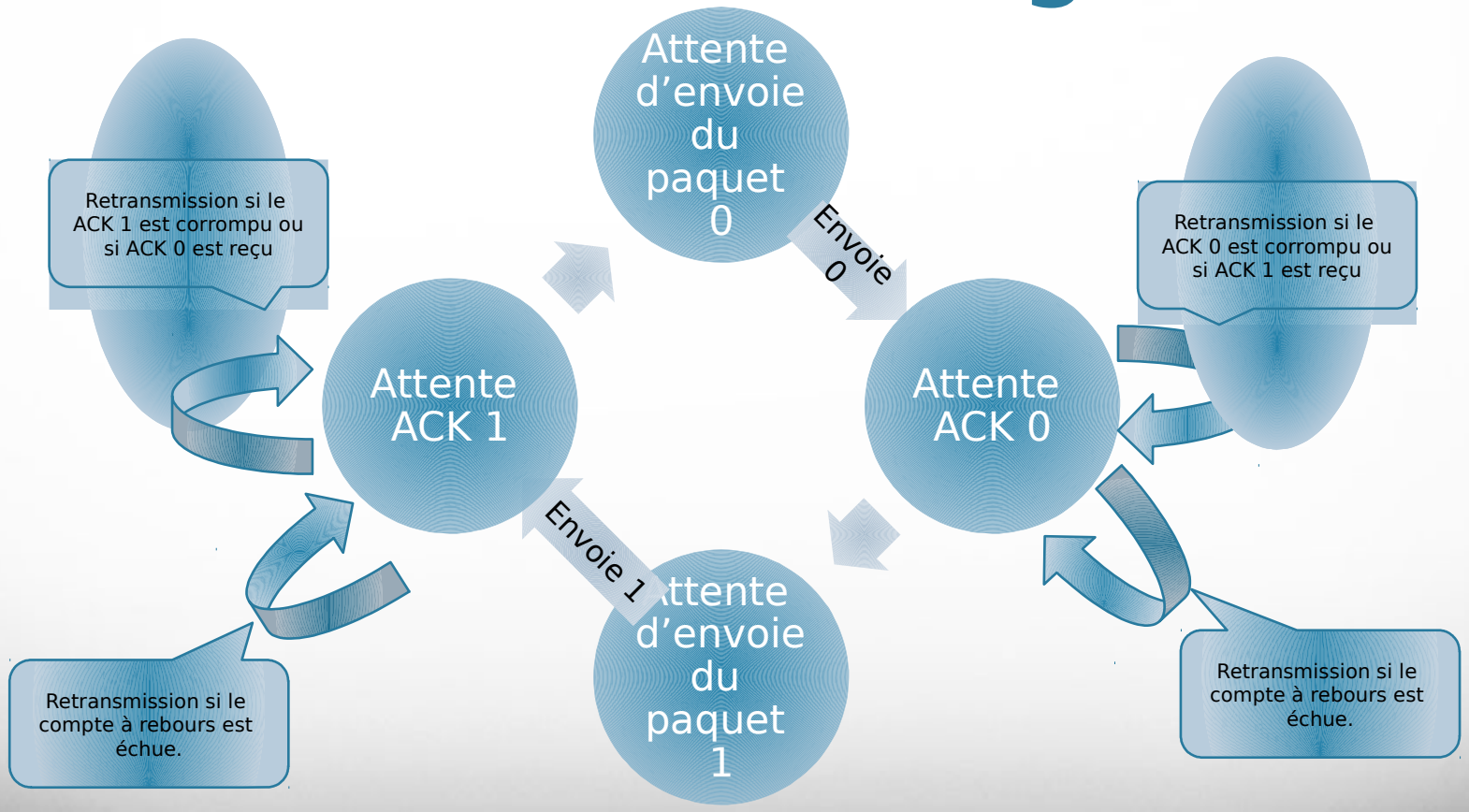
Les paquets perdues

- Bien que les trois derniers mécanismes fonctionnent bien ensemble dans le cas ou les paquets sont corrompus, ils assument toujours que les paquets se rendent...

Chronométrage

- Parce que la partie émettrice ne sait pas si le paquet envoyé fut reçu ou délayé ou que l'accusé réception fut lui aussi envoyé ou en retard, l'insertion d'un **chronométrage** agira en tant qu'arbitre pour retransmettre le paquet.
- Lorsque la partie réceptrice reste sans réponse pendant un certain temps, elle envoie à nouveau le message.
  - Le temps d'attente est déterminé par plusieurs facteurs….

Exemple de chronométrage



Performance

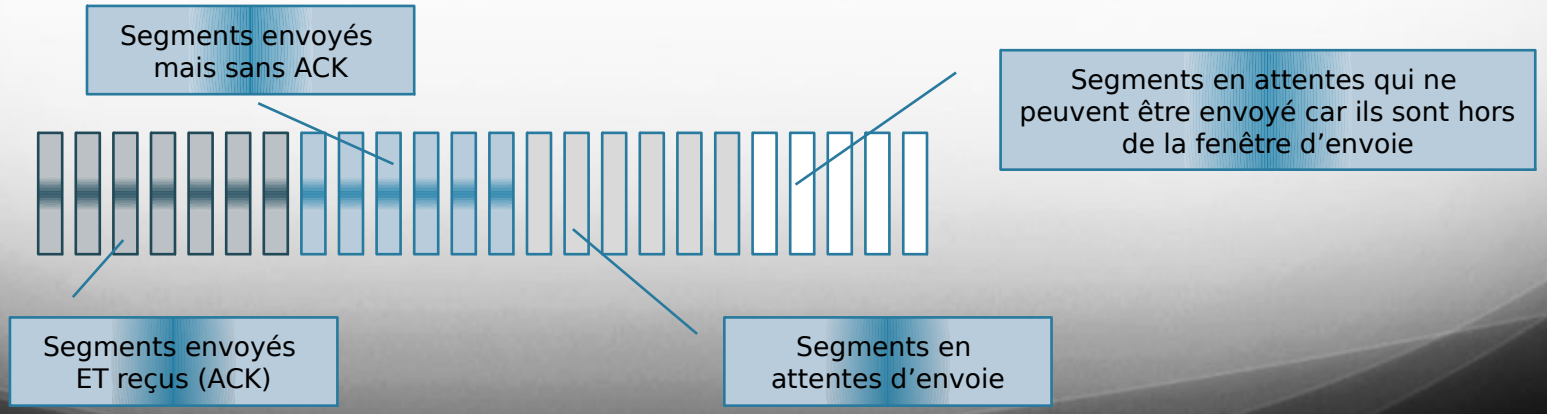
- Bien que le dernier modèle avec l'inclusion de chronomètre assure une fiabilité du transport des données, le problématique de la performance s'impose car:
  - L'application ne peut transmettre de données lorsque la couche transport est en 'attente' des accusés réception et que;
  - Pendant que le récepteur reçoit le message, le traite, retourne la réponse, le tube (la connexion) est complètement inutilisée.
- La solution : envoyé plus d'un paquet à la fois…

La fenêtre d’envoie

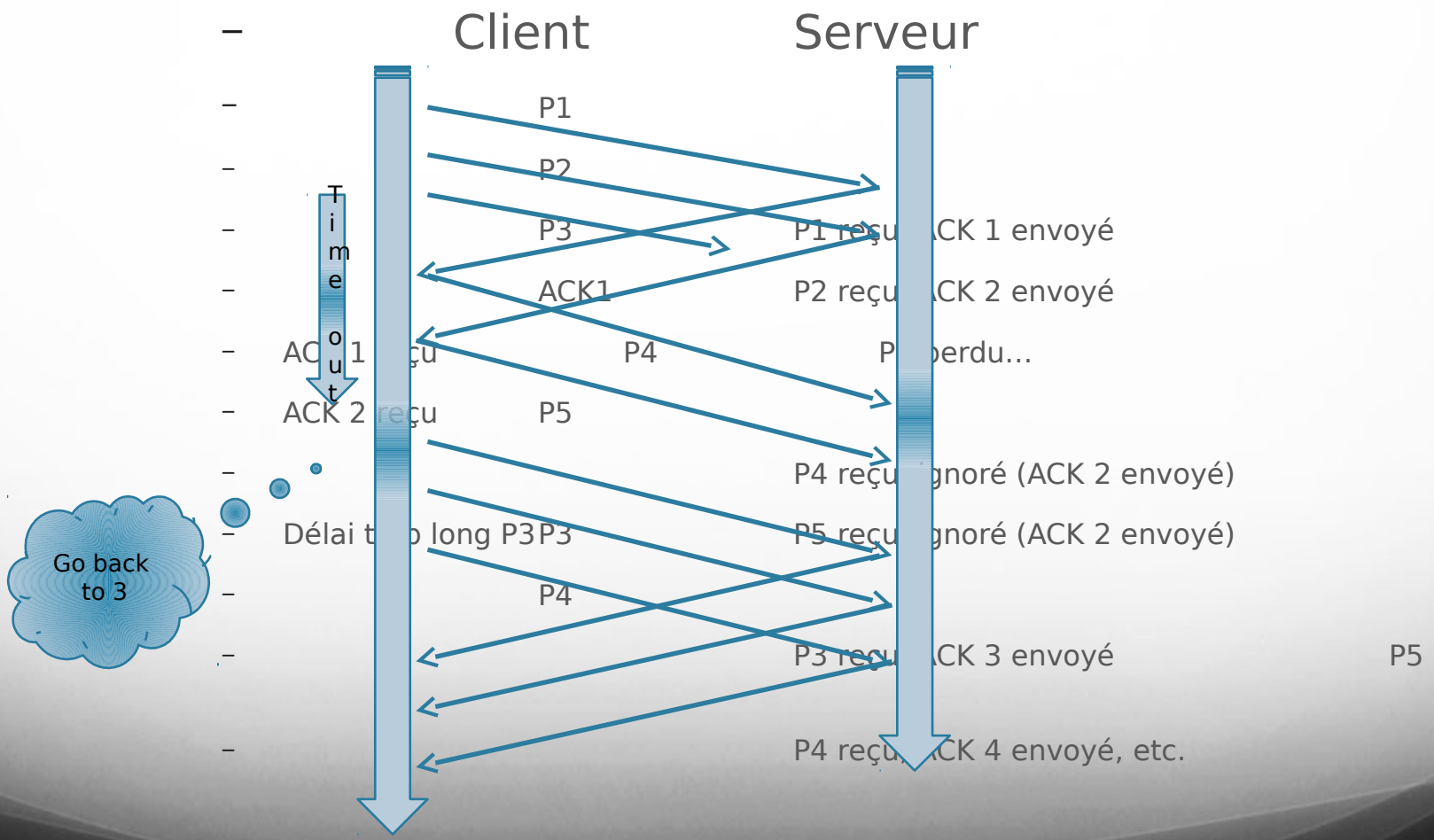
- Pour permettre à plusieurs message de prendre place dans le tube on introduit la notion de fenêtre d’envoie.
  - Au lieu d’envoyer un message et d’attendre son accusé réception, le client envoie plusieurs message et attends les accusées réception.
- Pour valider la transmission des données simultanées, deux modàles existe:
  - Go-Back-N (GBN)
  - Selective Repeat (SR)

Modèle GBN

- Le modèle GNB permet d'envoyer plusieurs segments à la fois jusqu'à un maximum de N segments.
- Une fenêtre d'envoie est donc créé et chaque segment dans cette fenêtre porte un numéro unique. Les segments en attentes sont placés dans un tampon.
- Lorsqu'un segment est reçu du coté du récepteur, il envoie un ACK avec le numéro du segment reçu.
- L'expéditeur attends le prochain ACK pour continuer l'envoie.



Modèle GBN


Client
Serveur
P1
P2
P3
ACK1
P4
P5
P3
P4
P5
Time out
Go back to 3
P1 reçu, ACK 1 envoyé
P2 reçu, ACK 2 envoyé
P4 perdu…
ACK 2 reçu
P4 reçu, ignoré (ACK 2 envoyé)
Délai trop long P3
P5 reçu, ignoré (ACK 2 envoyé)
P3 reçu, ACK 3 envoyé
P4 reçu, ACK 4 envoyé, etc.

Faiblesse du GBN

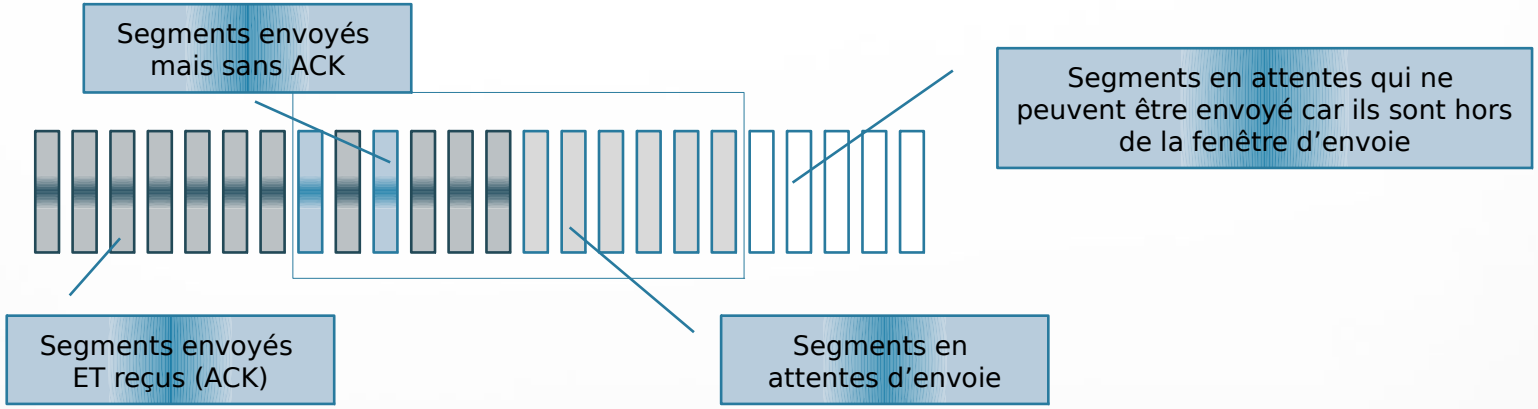
- La principale faiblesse de ce modèle est la retransmission de paquets qui peuvent être déjà rendus à destination.
  - Dans l'exemple, P4 et P5 furent discrédités malgré leurs bonnes réceptions.
  - ACK 2 est envoyé jusqu'à la réception de P3 (au cas ou ACK 2 n'était pas reçu)
- De plus, si ACK*n* est reçu et *n* fut déjà envoyé, l'expéditeur ignore cet ACK
- Finalement, si ACK*n* est reçu mais pas ACK*n-x*, le prochain paquet à envoyer sera P*n+1*

Modèle SR

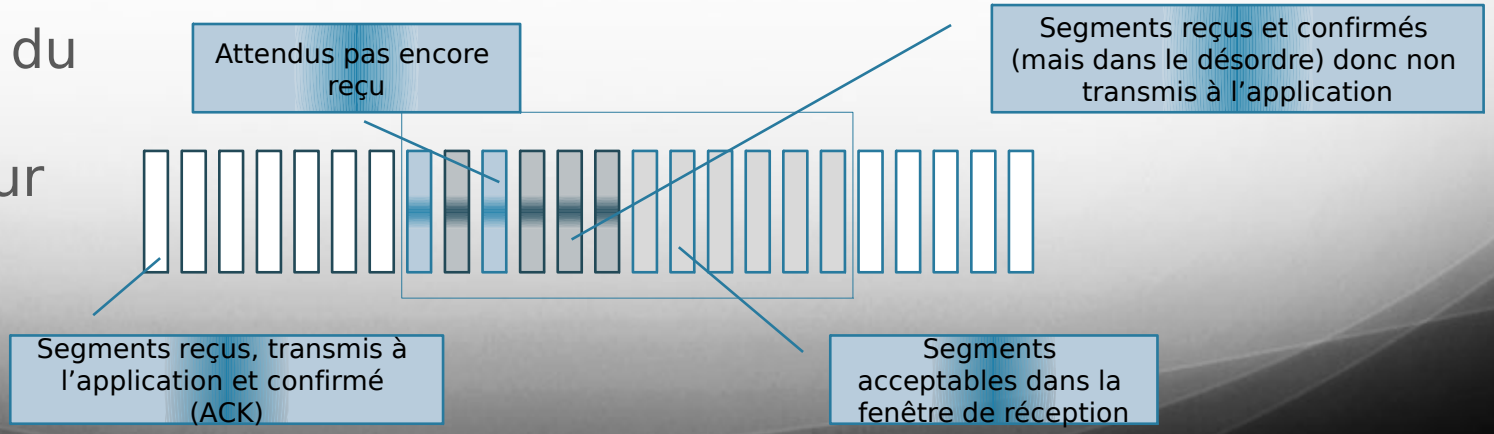
- Bien que le modèle GBN permet de remplir le tube, il possède le désavantage de refuser des paquets complet mais dans un mauvais ordre
  - Exemple : recevoir P4 avant P3. Ignore P4.
- Le modèle SR permet l'envoie 'sélectif' de paquets qui n'ont pas encore été reçus et confirmés.
- Ce modèle implique aussi un tampon du coté de la réception

Tampons SR

- Tampon du coté de l'expéditeur



- Tampon du coté du récepteur

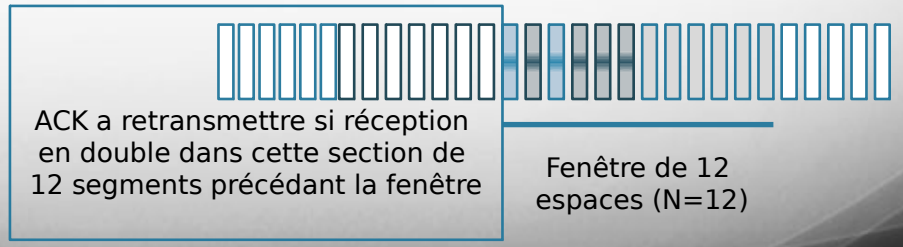


Événements et actions du coté de l'envoyeur

- Réception de donnée de la couche application:
  - Lorsque des données sont prêtes, l'expéditeur SR regarde le prochain numéro de séquence. Si le prochain numéro est dans la fenêtre, le segment est envoyé. Sinon, les données sont mises dans un tampon temporaire ou renvoyé à l'application.
- Le délai d'attente pour un ACK est trop long:
  - Étant donné que chaque segment possède son propre chronomètre, lorsque le délai expire, le paquet est tout simplement renvoyé.
- Un ACK est reçu:
  - Le segment est marqué comme étant confirmé. La fenêtre d'envoie est déplacée jusqu'au prochain segment qui attend sa confirmation ou jusqu'au prochain segment qui attend d'être envoyé. Si la fenêtre est déplacée les segments prêts à la transmission sont envoyés.

Événements et actions du coté de receveur

- Le segment reçu est à l'intérieur de la fenêtre
  - Envoyer un ACK pour ce paquet.
  - Si le paquet est nouveau l'insérer dans le tampon de la fenêtre. Sinon, l'ignorer.
  - Si le numéro du paquet est le plus petit envoyé ce paquet à l'application. Si d'autre paquets déjà reçus qui lui font suite, les envoyer à l'application.
  - Déplacer la fenêtre jusqu'au paquet non reçu.
- Le segment reçu est à l'extérieure de la fenêtre
  - Si le paquet est à l'intérieur de la position de la fenêtre – N jusqu'au coté gauche de la fenêtre: Envoyer un ACK
  - Sinon ignorer
- Le segment reçu est corrompu:
  - Ignorer.



À maitriser

- Les différentes méthodes de contrôles.
- Les deux types de fenêtre (GBN et SR)

Des questions?

Sources

- http://mathforum.org/library/drmath/view/54379.html
- Computer Networking, A top-down approach.