Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Admin Rés

Corrections TD n°4

QoS, TCP & Gestion de la congestion

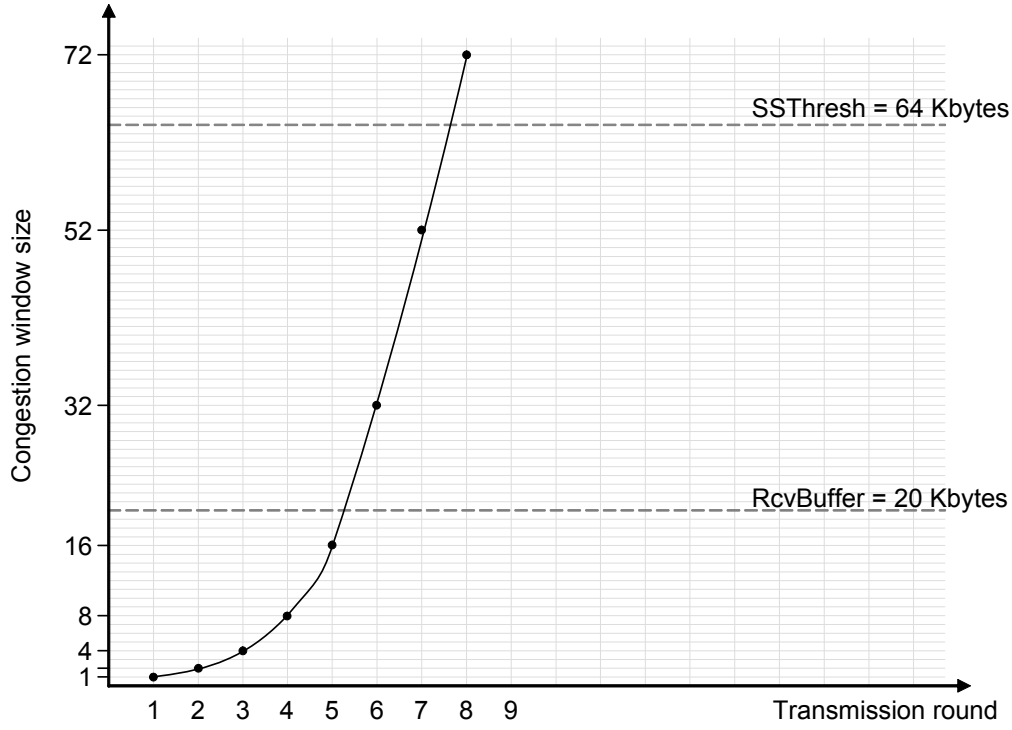
TCP & fenêtre de congestion

Soient deux hôtes A et B connectés à un même réseau local avec un RTT, « *round-trip time* », négligeable.
A envoie à B un gros volume de données en utilisant le protocole TCP :

- RcvBuffer = $20Ko$, le buffer de réception de B ;
- MSS = $1Ko$;
- le « ssthresh » = $64 * MSS$;

Il n'y a pas d'erreur de transmission, chaque hôte dispose d'un processeur rapide et les autres paramètres nécessaires sont standards.

a. Tracer l'évolution de la fenêtre de congestion durant le « *slow-start* » pour un réseau de $100Mb/s$:

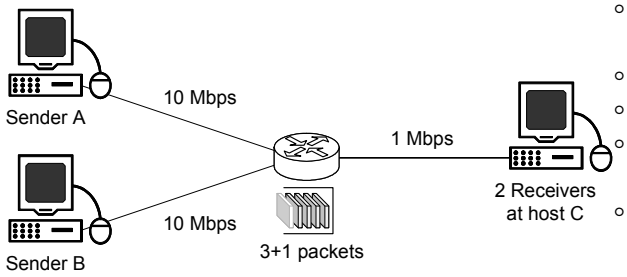


On note, tout d'abord, que les deux hôtes sont rapides et qu'il n'y a pas de pertes de segments : l'émetteur sera toujours autorisé à émettre une quantité de données égale à la taille maximale du buffer de réception, soit $20Ko$.
La fenêtre de congestion augmente au départ de façon exponentielle.
À l'étape 6, elle atteint une taille de $32 * MSS = 32Ko$ *supérieure à celle de la fenêtre de réception : l'émetteur n'est autorisé à émettre* $min\{CongWin, RcvWindow\} = 20$ *segments.*
Lorsque ces segments seront acquittés, la fenêtre de congestion ne sera que de $52 * MSS$ *au lieu des* $64 * MSS$ *attendus.*
Après, la fenêtre de congestion n'augmente toujours que de 20.

b. Comment évolue le tracé si le débit du réseau est réduit à $10Mb/s$? $1Mb/s$?
Le schéma reste le même quelque soit le débit, chaque « round » ou étape a une durée différente en fonction du débit.

c. À quel étape l'émetteur entre en phase d'évitement de congestion, « *congestion avoidance* » ?
À l'étape 8, la fenêtre de congestion atteint $72 * MSS$ *qui est supérieure au « ssthreshold » et l'émetteur passe en « congestion avoidance »*

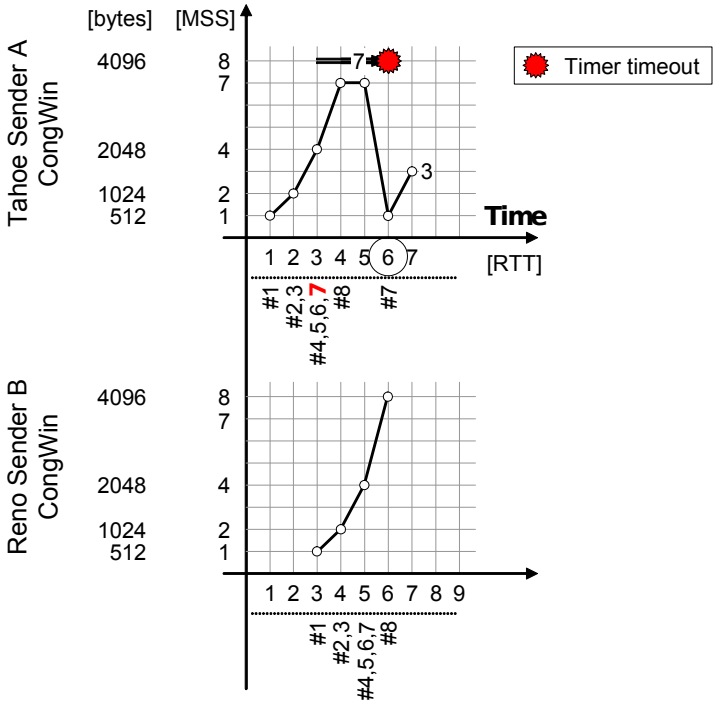
2 – Soit le réseau suivant :



- les hôtes A et B envoient par TCP $3,6Ko$ de données chacun, à l'hôte C ;
- la MTU est de $512octets$ pour tous les liens ;
- le $TimeoutInterval = 3 * RTT = 3 * 1sec$;
- le routeur dispose d'une capacité de 3 paquets en plus de celui actuellement transmis ;
- si le routeur doit détruire un paquet, il choisit le dernier paquet arrivé en provenance de l'hôte qui transmets le plus de paquets.

- le protocole TCP de l'hôte A exécute « TCP Tahoe » ;
- le protocole TCP de l'hôte b exécute « TCP Reno » ;
- l'hôte B commence sa transmission $2 * RTT$ après A.

a. Tracez l'évolution de la fenêtre de congestion sur les hôtes A et B.



On utilisera une grande fenêtre de réception, une transmission sans erreur sur tous les liens. Afin de simplifier le tracé, on considérera que tous les ACK arrivent exactement après un RTT et que la fenêtre de congestion est mise à jour à ce moment là.

Chaque hôte envoie $3,6Ko = 3687octets$, soient 7 paquets de $472octets$ et 1 paquet de $383octets$.

On ne tiendra pas compte de l'encapsulation des données (entête de trame, de datagramme IP, de segment TCP) ni des segments d'établissement de connexion dans cette proposition de correction.

Explication des étapes :

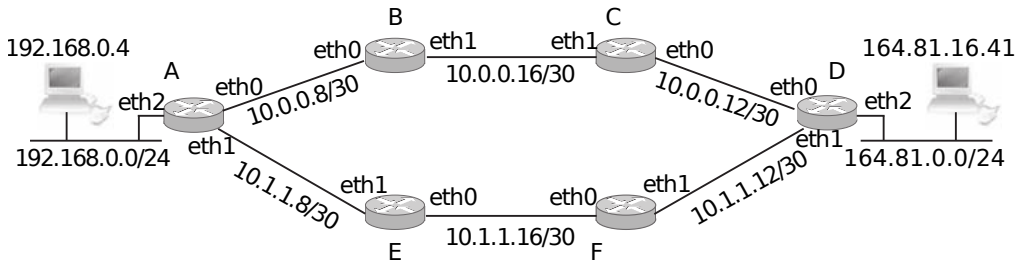
- *à l'étape 3, l'hôte A envoie 4 paquets en même temps que l'hôte B en envoie 1 : le routeur détruit le dernier paquet de l'hôte transmettant le plus, c-à-d le paquet n°7 de l'hôte A ;*
- *à l'étape 3, l'émetteur A utilise un seul compteur de retransmission RTO pour tous les paquets émis ;*
- *à l'étape 4, des acquittements sont reçus pour les paquets 4, 5, 6, mais pas pour le 7, le RTO continue et la fenêtre de congestion augmente de 3 et passe à $7 * MSS$;*
- *à l'étape 5, l'émetteur n'a plus rien à émettre, pas d'acquittement pour le paquet 7 ;*
- *à l'étape 6, le RTO arrive à zéro : réémission du paquet 7 et la fenêtre de congestion est réinitialisée à $1 * MSS$;*
- *à l'étape 7, un acquittement cumulatif des paquets 7 et 8 passe la fenêtre de congestion à $3 * MSS$.*

L'hôte B réalise ses envois sans problème et sa fenêtre de congestion augmente correctement.
Les algorithmes Reno et Tahoe dans ce cas là se comportent de la même façon, car on est en présence d'une congestion forte comme identifiée par Reno (RTO qui se déclenche) et non d'une duplication de ACK : étape 4, on acquitte en demandant le n°7, puis à l'étape 5 on acquitte de nouveau en demandant le n°7, et à l'étape 6 le RTO expire avant un $3^{ème}$ acquittement demandant le n°7 (3 acquittements dupliqués $\Longrightarrow$congestion faible).

Firewall & QoS

3 – D'après les connexions des différents routeurs :

a. Retrouvez le schéma du réseau d'interconnexion de ces différents routeurs :



b. La politique de sécurité

- empêcher les machines du réseau `192.168.0.0/24` de communiquer vers l'extérieur :
  - Sur le Routeur A :
  - si on veut bloquer la totalité des échanges en utilisant la « policy » :

```
xterm
# iptables -t filter -P FORWARD DROP
```

- les machines du réseau `192.168.0.0/24` sont autorisées à accéder à un serveur Web d'adresse `164.81.16.41` :

```
xterm
# iptables -t filter -A FORWARD  -m state --state ESTABLISHED,RELATED -j
ACCEPT
# iptables -t filter -A FORWARD -s 192.168.0.0/24 -d 164.81.16.41 -p tcp
--dport 80  -m state --state NEW -j ACCEPT
```

Avec le choix de la « policy » par défaut DROP, ces règles sont simplement ajoutées.

c. On veut empêcher l'accès SSH (port 22) depuis le réseau `164.81.0.0/24` sur le routeur D :

```
xterm
# iptables -t filter -A INPUT -s 164.81.0.0/24 -p tcp --dport 22 -j DROP
```

d. On veut limiter le trafic du protocole VNC à 10mbps :

Le protocole VNC est basé sur TCP (port 5900) et permet de visualiser l'écran d'un ordinateur à distance.

Sur quel routeur doit-on installer cette configuration de QoS, si l'accès par VNC se fait depuis une machine du réseau `192.168.0.0/24` vers une machine du réseau `164.81.0.0/24` (c-à-d qu'une machine du réseau `164.81.0.0/24` envoi régulièrement le contenu de son écran à une machine du réseau `192.168.0.0/24`) ?

Le trafic « volumineux » se fait d'une machine du réseau `164.81.0.0/24` vers une machine du réseau `192.168.0.0/24`, c'est donc lui qu'il faut limiter.
Pour le limiter, il faut utiliser le routeur le plus proche de l'émetteur, à savoir le routeur D.

Pour le choix de l'interface, il faut que ce soit l'interface de « sortie *» pour le trafic en provenance de `164.81.0.0/24`, c-à-d l'interface `eth0` ou l'interface `eth1` du routeur D.*
Ne connaissant pas l'interface choisi par le routage pour ateindre le réseau `192.168.0.0/24` depuis D, on peut mettre les règles de QoS sur les deux interfaces $\Longrightarrow$ *`IF=eth0` et `IF=eth1`.*

On applique sur D la configuration suivante pour réaliser de la QoS et pour chaque interface :

```
xterm
# tc qdisc add dev $IF root handle 1: htb default 10
# tc class add dev $IF parent 1: classid 1:10 htb rate 20mbps
# tc class add dev $IF parent 1: classid 1:20 htb rate 10mbps
```

Puis on mets en place la classification du trafic :

```
xterm
# iptables -t mangle -A PREROUTING -p tcp --sport 5900 -j MARK --set-mark 1
# tc filter add dev $IF protocol ip parent 1: handle 1 fw classid 1:20
```