

Algorithmes des réseaux

TP3 et 4 - TCP

Dans ce TP nous allons déployer un réseau IPv4 comportant un goulot d'étranglement afin de tester l'efficacité des différentes variantes du protocole TCP. Pour la suite du TP, vous aurez besoin de deux postes et deux commutateurs (**Prenez un Cisco et un Juniper**).

1 Réinitialisation des équipements

Il vous est demandé de réinitialiser les commutateurs **À LA FIN DE LA SÉANCE**.

1.1 Cisco

Pour les équipements cisco, placez vous en mode **monitoring** et entrez les commandes suivantes :

```
delete flash:vlan.dat
write erase
reload
```

Si un message vous demande si vous souhaitez sauvegarder la configuration, refusez.

```
System configuration has been modified. Save? [yes/no]: no
```

Après le redémarrage, le routeur vous proposera de charger une configuration automatique. Refusez en répondant No à chacune des trois questions.

1.2 Juniper

Pour les juniper :

```
> request system zeroize
warning: System will be rebooted and may not boot without configuration
Erase all data, including configuration and log files? [yes,no] (no) yes
```

Attention à ne pas éteindre les équipements lors de la procédure.

2 Configuration TCP

Vérifiez quelles versions de TCP sont disponibles directement sur votre poste, via la commande :

```
sysctl net.ipv4.tcp_available_congestion_control
```

Pour sélectionner la version de TCP à utiliser, vous pouvez utiliser la commande (en tant qu'administrateur) :

```
sysctl -w net.ipv4.tcp_congestion_control=version
```

Configurez vos postes pour qu'ils utilisent la version **TCP Reno**.

Certaines versions de TCP peuvent être disponibles en module et doivent au préalable être activées (i.e., il faut charger le module correspondant) avant de pouvoir les utiliser. Les modules en question se trouvent dans le répertoire :

```
ls /lib/modules/`uname -r`/kernel/net/ipv4/
```

Ils peuvent ensuite être activés en chargeant le module via modprobe :

```
modprobe tcp_bic (ou tcp_yeah, tcp_bic...)
```

Pour générer du trafic TCP, vous devez installer le logiciel **iperf3**. Pour tracer la courbe des résultats, vous pouvez installer le logiciel **xplot** (via apt install).

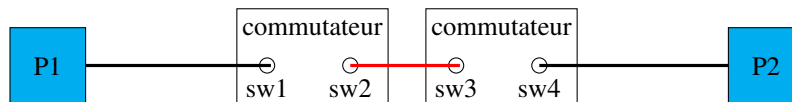


FIGURE 1 – Topologie du réseau

3 Mise en place du réseau

La première étape de l'exercice consiste à déployer le réseau. Pour ce faire, vous allez utiliser 2 postes et un commutateur afin de réaliser la configuration illustrée sur la Fig 1.

Un premier poste (PC1) servira de générateur de flux. Il est connecté sur le port sw1 d'un premier commutateur. Ce commutateur est lui-même relié via son port sw2 au port sw3 d'un second commutateur, auquel est connecté le deuxième poste PC2 (sur le port sw4). PC2 fera office de receveur. De ce fait, les flux iront de PC1 à PC2 en passant par l'interface du PC1, sw1, sw2, sw3, sw4 et l'interface du PC2. **Assurez vous de prendre un commutateur Cisco et un commutateur Juniper.** Après le déploiement physique du réseau réalisez les opérations suivantes :

3.1 Poste

1. Identifiez l'interface vous connectant à l'armoire contenant les commutateurs.
2. Par défaut, cette interface est down. Activez la à l'aide de `ifconfig` ou `ip`.
3. Configurez une adresse IP sur ces interfaces (interface PC1 et PC2) de manière à ce que les deux postes soient dans le même sous-réseau.
4. Vérifiez la connectivité IP dans les deux sens.

3.2 Bridage la connection

On cherche à brider le lien entre les commutateurs, afin que le débit assuré par celui-ci soit réduit (lien rouge sur la Fig. 1).

1. Limitez le débit du lien à 10Mb/s sur le commutateur.
 Sur Cisco, utilisez la commande `speed` sur les interfaces concernées pour les passer en 10Mb/s (`enable, conf t, int [ifname], puis la commande speed`). Réduisez encore davantage le débit à l'aide de la commande `srr-queue bandwidth limit` pour atteindre 20% du débit disponible (2Mb/s).
 Sur Juniper, utilisez également la commande `speed (set interfaces [ifname] speed 10m)`;
2. Précisez sur le commutateur que les liens sont full-duplex.

4 Analyse de performance

Pour analyser les performances de TCP, nous pourrions utiliser Wireshark mais ce dernier ne permet pas de récupérer des informations importantes telles que la taille de la fenêtre de congestion. Cependant, **iperf3** peut afficher la taille de la fenêtre de congestion sur l'émetteur.

Nous allons maintenant réaliser notre première mesure. Pour ce faire, effectuez les opérations suivantes.

1. Lancez la partie serveur d'**iperf3** sur le poste PC2
2. Dans un nouveau terminal, lancez la partie cliente d'**iperf3** sur PC1 en indiquant le port du serveur. Effectuez un envoi d'une durée de 30s avec une intervalle de mesure de 0.5s.
3. Quel est le débit atteint par **iperf3** ?
4. Comment évolue `cwnd` ? Quel est l'effet d'une retransmission (`retr`) sur `cwnd`.
5. Nous allons maintenant tracer la variation de la fenêtre de congestion au cours du temps. Pour ce faire, vous pouvez utiliser l'outil de votre choix (excel, gnuplot...). L'output d'**iperf** commence par un header de 3 lignes, et finit par un footer de 6 lignes. Entre les deux se trouvent $f \times t$ mesures, avec f le nombre de mesure par seconde et t le temps de l'expérience. Une fois ces lignes supprimées, **xplot** peut afficher la courbe décrite par les données.

Une manière simple d'obtenir cette courbe est donc d'utiliser la commande suivante :

```
iperf3 -c <ip add> -p <port> -i 0.1 -t 30 | rev | cut -f9 -d ' ' | rev | head -n (f * t + 3) | tail -n (f * t) | xplot.
```

(Vous allez devoir installer **xplot**)

Si la courbe vous paraît bizarre, utiliser les curseurs pour changer l'échelle du graphe. Les phases de TCP devraient alors devenir visibles.

6. Sur le graphique obtenu, identifiez les différentes phases de TCP (slow-start, évitement de congestion, détection de pertes, etc.). Quel est la valeur initiale de la fenêtre de congestion ? Pourquoi ?
7. Nous allons maintenant jouer sur le délai pour augmenter le RTT afin de simuler un réseau dit *Long Flat Network* (LFN, cf. RFC 1072). En vous basant sur le RFC 1072, modifier l'interface de PC1 afin de créer un LFN à l'aide des commandes suivantes :

```
tc qdisc add dev ifname root netem delay Xms # ajoute Xms de délai
tc qdisc show                               # visualiser la config. de tc
tc qdisc del dev ifname root netem          # supp. la config. d'une interface
```

8. Analysez le débit atteint par TCP Reno sur un tel réseau. Analysez également la variation de la fenêtre de congestion. Que constatez-vous ? Pourquoi ?
9. N'y a-t-il pas une version de TCP plus adaptée aux réseaux LFN ? Si oui, configurez vos postes pour utiliser cette version de TCP et relancez les tests de performance. Que constatez-vous ?