M3102 : TP2 sur Network Simulator ns2

Ce qu'on va faire dans ce TP : Analyser les performances obtenues par des stations en fonction

- de leur version de TCP
- des autres stations présentes
- de la politique de gestion de buffer (politique d'abandon de paquets)
- de la gestion des classes de service

Pourquoi : Pour comprendre l'impact de ces différents paramètres sur les performances du réseau (débits et délais obtenus par les stations, niveau et durée de la congestion)

Comment : Par l'outil de simulation ns2, permettant d'implémenter ces différents algorithmes (versions de TCP, gestion de buffers, de files et de classes de trafic).

Description de ns2 :

Le simulateur ns est basé sur deux langages : un simulateur écrit dans le langage de programmation orientée objet C++ (langage de POO comme Java), et sur un interpréteur OTcl (une extension orientée objet de Tcl – *Tool Command Language*), utilisé pour exécuter des scripts de commandes de l'utilisateur.

Dans ce TD vous n'allez pas coder en OTcl, mais simplement utiliser du code déjà écrit pour simuler des scénarios intéressants de réseaux étendus vus en cours, en interpréter les résultats et ainsi assimiler les fonctionnements des protocoles utilisés en constatant leur intérêt en terme de performance.

Source : T. Jimenez and A. Altman, *NS Simulator Course for Beginners*.

Actions préalables à effectuer depuis votre compte :

- Sous Ubuntu, loggez-vous sur votre compte (un des 2 membres du binôme) .
- Ouvrir un terminal et taper : `ns-config`
- Télécharger le répertoire compressé *Codes_ns2_etudiant.zip* depuis http://www.i3s.unice.fr/~sassatelli/M3102/
- Décompresser le fichier
- Ouvrez un terminal, et placez-vous dans le répertoire *Codes_ns2_etudiant* (rappel : par la commande shell `cd <nom_repertoire>`)

L'environnement de simulation :

- Dans le répertoire *Codes_ns2_etudiant*, vous devez voir (rappel : par la commande shell `ls`) plusieurs fichiers .tcl : *ex1.tcl, ex3.tcl, droptail.tcl, red.tcl, diffserv.tcl.*
- Vous devez également y trouver des fichiers écrits en langage Perl (*Practical Extraction and Report Language*), ce langage interprété est notamment utilisé pour filtrer et traiter des fichiers de données ASCII

facilement en Unix : *throughput.pl* et *throughput_multi.pl*. Ces scripts vont traiter les fichiers texte générés par le simulateur, pour en extraire les quantités intéressantes : débit, taille de la fenêtre de congestion de TCP, taux de perte de paquet, taille de la file d'attente à un routeur, etc.

- Ces scripts généreront d'autres fichiers eux directement lisibles par un outil de création de graphiques, tel Gnuplot sous Unix.

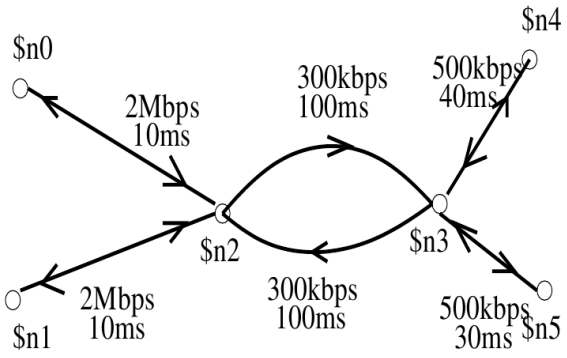
- 1 compte-rendu par étudiant·e
- Vous prenez le fichier docx depuis http://www.i3s.unice.fr/~sassatelli/M3102/ . Vous l'ouvrez avec Libroffice et vous faîtes le compte-rendu à partir de là (comme en TP1). Vous collez les figures obtenues dans ce fichiers, et vous les commentez , pour répondre à chaque question.
- Pour toutes les commandes à entrer dans le terminal, vous copiez –collez depuis le docx.

I Simulation et étude de TCP

I.1. Une connexion TCP et une connexion UDP

Description de la configuration réseau étudiée dans cette partie :

Nous considérons la topologie ci-dessous (nœuds et liens). 2 flows sont générés : une application FTP générant du trafic depuis n0 vers n4, et une application de type CBR (Constant Bit Rate), telle que de la VoIP, de n1 vers n5. On rappelle que FTP est encapsulé dans TCP (puis dans IP), et CBR par UDP.



I.1.1. Avec TCP Tahoe

Ouvrez le fichier ex1.tcl dans un éditeur de texte : `gedit ex1.tcl &`

La ligne 60 doit indiquer (ou changez-la pour qu'elle indique) : `set tcp [new Agent/TCP]`

Taper dans un terminal :

```
> ns ex1.tcl
> perl throughput.pl out.tr 1 ftput floss
> gnuplot
```

```
> plot "WinFile" using 1:2 title 'Cwnd Flow 1' with lines linetype 1, "floss"
using 1:2 title 'Loss rate' with lines linetype 2, "QueueSize" using 1:2 title
'Queue size at n2' with lines linetype 3
```

Et dans un autre terminal :

```
> gnuplot
> plot "ftput" using 1:2 title 'Rate Flow 1' with lines linetype 1, "ftput"
using 1:3 title 'Rate Flow 2' with lines linetype 2
```

1 Observer le réseau avec Nam : Network Animator.
   a Que voyez-vous ? Identifiez-vous les paquets TCP et les ACKs ?
   b Que constatez-vous quant à l'évolution du nombre de paquets TCP envoyés par fenêtre de congestion ?
   c Quel phénomène apparaît à partir de 1.7s ? Pour quelle raison ?
   d Quel phénomène apparaît à partir de 2.8s ? Pour quelle raison ?

2 Observer les courbes obtenues à partir de gnuplot.
   a Quel est le débit D_1 recherché par la connexion TCP ? Faire le lien avec la capacité du goulot d'étranglement (*bottleneck link*) et le débit D_2 constant de la connexion CBR portée par UDP. Formaliser la réponse en exprimant les bandes passantes disponibles des liens ($Bw_{dispo,lien\ i}$) et des chemins ($Bw_{dispo,path\ j}$).
   b Expliquez l'évolution de la fenêtre de congestion en faisant le lien avec la taille de la file d'attente au nœud 2 (commentez l'évolution de la file d'attente).
   c Zoomez de façon à identifier les 2 phases spécifiques (vues en cours) de croissance de la fenêtre de congestion. On y définit 3 valeurs spécifiques de la façon suivante:
      - $cwnd_1$: valeur de cwnd quand l'évolution de la fenêtre passe d'exponentielle à linéraire
      - $cwnd_2$: valeur max de cwnd telle que pas de paquets mis en file
      - $cwnd_3$: valeur de cwnd au moment au la source TCP détecte la une perte

      Pour chacune, exprimer la formule littérale et détailler l'application numérique, dont les valeurs trouvées doivent correspondre à celles que vous lisez sur la figure.
   d En quoi voyez-vous la spécificité de TCP Tahoe par rapport aux autres versions de TCP ?
   e Que se passe t-il à t=80s ?

Tapez depuis gnuplot pour en sortir :

```
> exit
```

I.1.2. Avec TCP Reno

Changer la ligne 60 du fichier ex1.tcl : set tcp [new Agent/TCP/Reno]
Reprenez les questions 2.b et 2.d du I.1.1.

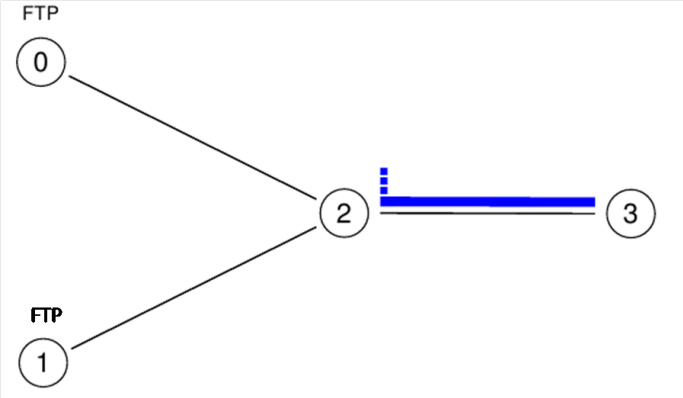
I.1.3. Avec TCP New Reno

Changer la ligne 60 du fichier ex1.tcl : set tcp [new Agent/TCP/Newreno]
Reprenez les questions 2.b et 2.d du I.1.1.

I.2. Plusieurs connexions TCP

Description de la configuration réseau étudiée dans cette partie :

Nous considérons la topologie ci-dessous (nœuds et liens). 2 flows sont générés par 2 applications FTP générant du trafic de n0 vers n3 et n1 vers n3. Nous étudions le partage de la capacité du lien goulot d'étranglement par ces 2 connexions TCP, en fonction de la version de TCP dans le système correspondant aux 2 machines émettrices n0 et n1.



Ouvrez le fichier ex3.tcl dans un éditeur de texte : `gedit ex3.tcl &`
Pour les versions de TCP, vous changerez les lignes 83et 85 en fonction des versions de TCP à tester ensemble : `set tcp [new Agent/TCP]` pour TCP Tahoe, `set tcp [new Agent/TCP/Reno]` pour TCP Reno, `set tcp [new Agent/TCP/Newreno]` pour TCP New Reno.
Pour changer les délais subis par chacune des connexions, vous changerez les lignes 62 et 63.
Taper dans un terminal :

```
> ns ex3.tcl
> perl throughput_multi.pl out.tr 1 ftput floss
> gnuplot
> plot "win" using 1:2 title 'Cwnd Flow 1' with lines linetype 1, "win" using
1:3 title 'Cwnd Flow 2' with lines linetype 2, "floss" using 1:2 title 'Loss
rate' with lines linetype 3, "QueueSize" using 1:2 title 'Queue size at n2'
with lines linetype 4
```

Et dans un autre terminal :

```
> gnuplot
> plot "ftput" using 1:2 title 'Rate Flow 1' with lines linetype 1, "ftput"
using 1:3 title 'Rate Flow 2' with lines linetype 2
```

I.2.1. Equité entre les différentes versions de TCP (en conditions symétriques)

1. Avec 2 connexions TCP Tahoe : quelles fractions de bande passante les 2 connexions obtiennent-elles?
2. Quelle devrait être la taille du buffer de n2 (en nombre de paquets) pour éviter toute perte (le résultat dépend de la fenêtre maximum de congestion possible, habituellement fixée par défaut à 64 MSS).
3. Tahoe vs Reno : quelle connexion obtient le plus de bande-passante, et pourquoi ? Répondre en analysant les mécanismes de reprise sur perte des 2 versions.
4. Reno vs New Reno : même question en modifiant les délais entre sources et n2 à 10ms.

I.2.2. Equité entre des sessions TCP avec des RTT différents

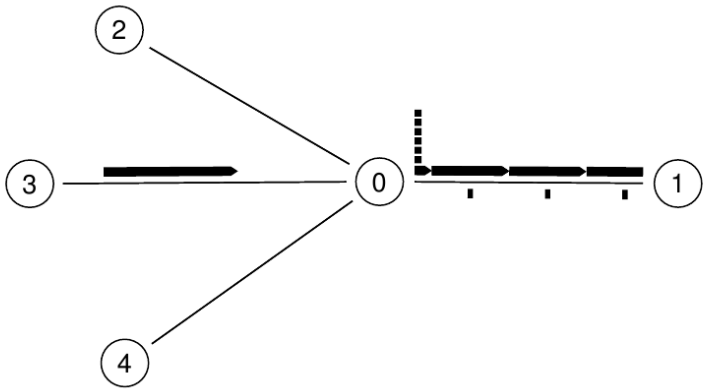
1 On considère 2 connexions TCP Reno Vérifiez la symétrie de performance des 2 connexions quand le délai entre la source et n2 est à 10ms pour la connexion 1, et 10ms pour la connexion 2.

2 Mettre le délai entre la source et n2 à 100ms pour la connexion 1, et 1ms pour la connexion 2. Comment expliquez-vous la différence de performance : formulez la raison en réfléchissant au temps de réaction que va pouvoir avoir TCP pour chaque connexion.

II Buffer management et QoS : cas de RED et implémentation de DiffServ

II.1. Buffer management: Random Early Discard (RED)

Nous allons analyser 2 techniques de gestion de buffer vues en cours: Droptail et RED. La topologie considérée est 3 sessions TCP (toujours avec une application FTP), de 2, 3 et 4 vers 1. La taille maximale du buffer au nœud 0 est initialement de 100 paquets, et le goulot d'étranglement est le lien 0-1 qui a une capacité de 3 Mbps.



Ouvrez le fichier droptail.tcl et tapez dans un terminal :

```
> ns droptail.tcl
> perl throughput_DTRED.pl out.tr 1 ftput
```

1 Visualisez le débit total utilisé sur le lien entre 0 et 1, et commentez le taux d'utilisation :

```
> gnuplot
> plot "ftput" using 1:2 title 'Total rate' with lines linetype 1
```

2 Visualisez ensuite l'évolution de la taille de la file d'attente au nœud 0 (dans un autre terminal:

```
> gnuplot
> plot "./QueueSize" using 1:2 title 'Queue size with droptail' with
lines linetype 1
```

ainsi que le délai de traversée des paquets de leur source à leur destination (dans un autre terminal) :

```
> perl delay.pl out.tr
```

Commentez l'impact de la file d'attente sur le délai subi par les paquets.

3 Une solution pour réduire ce délai est de réduire la taille du buffer : ligne 61 de droptail.tcl, changez 100 en 5. Relancez toutes les commandes pour tracer la nouvelle évolution de la file d'attente, le délai subi. Que constatez-vous ? Exprimer le temps passé en file par un paquet si la file contient *q* paquets.

4 Vérifiez maintenant si le taux d'utilisation du lien entre les 2 routeurs s'est maintenu :

```
> gnuplot
> plot "ftput" using 1:2 title 'Total rate' with lines linetype 1
```

Que constatez-vous ?

5 Expliquez ce phénomène en examinant les fenêtres de congestion :

```
> gnuplot
> plot "win" using 1:2 title 'Cwnd Flow 1' with lines linetype 1, "win"
using 1:3 title 'Cwnd Flow 2' with lines linetype 2, "win" using 1:4
title 'Cwnd Flow 3' with lines linetype 3
```

expliquer la cause du manque d'utilisation.

Ouvrez le fichier red.tcl et tapez dans un nouveau terminal :

```
> ns red.tcl
> perl throughput_DTRED.pl out.tr 1 ftput
```

6 Quelle est la plage de valeurs dans laquelle varie la taille de la file d'attente au nœud 0 ?

```
> gnuplot
> plot "./ave.tr" using 2:3 title 'Average queue size with red' with
lines linetype 1, "./cur.tr" using 2:3 title 'Current queue size with
red' with lines linetype 2
```

Que pensez-vous que cela va avoir comme impact sur le temps de transit des paquets entre la source et la destination ? Le vérifier grâce au résultat donné par :

```
> perl delay.pl out.tr
```

7 Vérifiez maintenant si le taux d'utilisation du lien entre les 2 routeurs s'est maintenu :

```
> gnuplot
> plot "ftput" using 1:2 title 'Total rate' with lines linetype 1
```

Que constatez-vous ?

8 Expliquez ce phénomène en examinant les fenêtres de congestion :

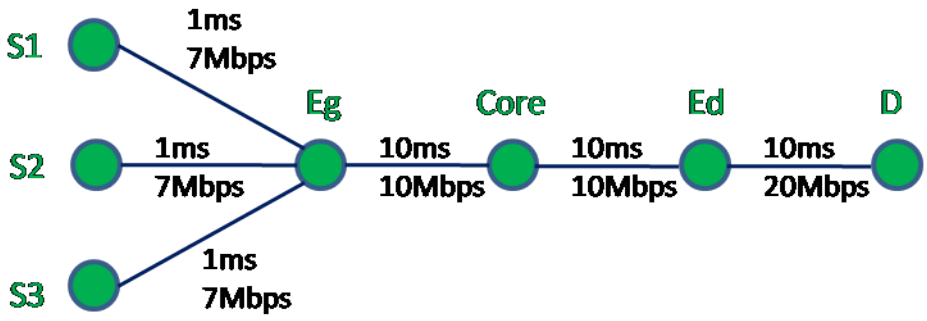
```
> gnuplot
> plot "win" using 1:2 title 'Cwnd Flow 1' with lines linetype 1, "win"
using 1:3 title 'Cwnd Flow 2' with lines linetype 2, "win" using 1:4
title 'Cwnd Flow 3' with lines linetype 3
```

expliquer ce qui a changé par rapport au cas avec droptail.

9 Faire varier les variables thres_ et maxthres_ (lignes 23 et 24), par exemple en les mettant à 2 et 5 respectivement, et observer la taille de la file d'attente résultante.

II.2. Implémentation de DiffServ pour la QoS au niveau IP: policing (application des priorités), buffer management (gestion de buffer) et scheduling (ordonnancement)

Nous considérons maintenant l'utilisation de DiffServ, qui (cf. cours) est un mécanisme qui permet de marquer les paquets IP (dans le champ DS de l'entête IP) pour leur attribuer une certaine priorité. Ces paquets seront ensuite traités selon leur priorité (ils seront envoyés avant d'autres arrivés plus tôt dans un routeur, ou auront une probabilité plus ou moins forte d'être abandonnés dans les routeurs du cœur du réseau.
Comme dans les précédents exercices, nous allons considérer une certaine topologie de réseau avec du trafic prédéfini. La figure ci-dessous représente la topologie. Trois flows UDP portant une application CBR de débit 10Mbps, sont générés à S1, S2 et S3, à destination de D.
Les paquets IP générés à S1, S2 et S3 vont d'abord arriver au routeur Eg, routeur de bord de l'opérateur, où ils vont être marqués avec des priorités différentes, et traités en conséquence dans les routeurs Eg, Core et Ed. Ed fait de même avec le trafic dans l'autre sens venant de D (seulement les ACKs de possibles connexions TCP dans notre cas).



1 Ouvrir le fichier diffserv.tcl. Lisez et analysez ce fichier depuis l'annexe (à la fin de ce document) grâce aux commentaires :
   - 1.a Sur quel critère les 3 flows sont-ils différenciés ? (type d'application ? ou paire source-destination ? ou protocole de transport ?...)
   - 1.b Quel est le lien entre nombre de files physiques et virtuelles, nombre et nature des critères ?
   - 1.c Décrivez le type de buffer management associé à chaque file physique.
   - 1.d Quels sont les choix possibles, dans le code tel qu'il vous est fourni, pour l'ordonnancement entre les différentes files physiques ?
   - 1.e D'après vous, à quoi sert la variable meanPktSize, et à quoi correspondent les valeurs qu'elle prend ?

2 Assurez-vous que la variable selecSched est à 1 et les poids de la politique Weighted Round Robin à 10, 5 et 2 respectivement et exécutez :

```
> ns diffserv.tcl
> perl throughput_multiDS.pl out.tr 1 ftput floss
> gnuplot
> plot "ftput" using 1:2 title 'Flow 1' with lines linetype 1, "ftput" using 1:3 title 'Flow 2' with lines linetype 2, "ftput" using 1:4 title 'Flow 3' with lines linetype 3
```

1.a Comment expliquez-vous les débits obtenus par chaque flow ? Discutez les valeurs de débits obtenus.

1.b Pour obtenir le délai moyen de transfert des paquets de chaque flow, exécutez exécutez (dans un autre terminal et même répertoire) :

```
perl delay_multi.pl out.tr
```

Que pouvez-vous dire ?

3 Changez les poids de la politique Weighted Round Robin à 19, 2 et 1 respectivement. Relancez les commandes ci-dessus et répondez de nouveau à la question a.

Source

Tania Jimenez et Eitan Altma, *Course ns2 for beginners*, online