

## NS2 - Simulations Avancées & Développement

---

*Objectif* : Maîtriser un simulateur réseau

*Notions* : Parsing input/output, analyses et interprétations de résultats à grande échelle, modifications OTCL/C++ dans NS2

---

### Manipulations automatisées de NS2 ( $\approx 18$ pts - retours individuels)

#### **Exercice 1 : Congestions et flux TCP ( $\approx 6$ pts - à rendre pour le Vendredi 17 Novembre)**

**Q 1.** Finissez les deux exercices du TP précédent. Fournissez vos scripts de configuration TCL (et/ou ceux vous permettant de générer les scripts TCL pour la topologie à 104 sommets).

**Q 2.** Rédigez une synthèse des résultats obtenus en cinq pages maximum (hors annexes de configuration et vos quelques figures - chacune devant être commenté par une demi page de texte en moyenne). Cette synthèse décrira d'abord de manière précise votre environnement de simulation (les paramètres réseaux utilisés et important pour la couche transport). Puis il s'agira essentiellement d'analyser, de comparer et de commenter le comportement de vos simulations au niveau des liens/files d'attente (vue opérateur) comme au niveau des flux TCP (vue applicative pour les utilisateurs). Mettez en évidence les limites de vos choix de simulation (du côté opérateur comme du côté applicatif). Les bonnes questions à se poser sont les suivantes : est-ce que le partage de ressources est équitable ? comment définir la taille d'une bonne file d'attente ? est-ce qu'il existe des versions de TCP plus performantes que d'autres dans cet environnement ?

#### **Exercice 2 : Topologie et trafic réaliste ( $\approx 12$ pts - à rendre pour le Vendredi 22 Décembre)**

**Q 1.** Sur base des fichiers topo.top et trafic.traf que votre aimable encadrant de TP vous aura fourni, mettez en oeuvre une simulation réaliste (5 minutes simulées). Chaque volume donné dans la matrice de trafic devra être subdivisé en un ensemble de sous flux TCP dont les dates de départ sont également à déterminer. Pour cela vous utiliserez la (ou les) distribution(s) qui vous semble(nt) les plus appropriée(s) pour générer un trafic pseudo-réaliste. Une autre approche (facultative si utilisée pour du trafic de fond en plus ou alternative si utilisée comme trafic principal avec une poignée de flux TCP "témoins" par dessus) consiste à utiliser le modèle ON/OFF étudié en cours.

Fournissez vos scripts de configuration (pas nécessairement le produit TCL) et justifiez en commentaires vos choix réseaux (distributions, taille/type des files, type de flux et configuration TCP, etc) en terme de réalisme.

**Q 2.** Analysez la charge et le taux de perte du ou des "pires liens" sur l'ensemble de la simulation (les trois liens sur lesquels vous observez le plus de pertes par exemple). Faites en de même pour les pires flux (ceux qui subissent le plus de re-transmissions). Commentez et analysez en profondeur les résultats obtenus (par exemple, la charge des liens et le débit utile de vos flux) et le type de trafic engendré. Pensez à évaluer votre trafic à différentes échelles de temps (2-3 pages de rapport hors figures) pour mieux comprendre la nature des congestions.

**Q 3.** A priori, le réseau qui vous est fourni est sur-dimensionné par rapport au trafic à injecter, trouvez un moyen de le mettre à “l’épreuve de manière réaliste” en intensifiant son activité. Utilisez quelques flux TCP témoins pour illustrer les différences avec le scénario précédent (1-2 pages de rapport hors figures).

**Q 4.** Modifiez la taille et le type des files (par exemple RED, DRR, etc) ainsi que les émetteurs/récepteurs TCP (Tahoe, New Reno, Sack, Delayed Ack, configuration des seuils TCP, etc) avec pour objectif, si possible, d’améliorer les performances de votre réseau et surtout celles de vos flux. Justifiez vos critères d’évaluation pour la comparaison (testez plusieurs combinaisons). Expliquez et analysez les nouveaux résultats obtenus dans un rapport de 2-3 pages.

## Développement NS2 : ECMP ( $\approx 2+$ pts - retours par groupe de quatre étudiants maximum)

Par défaut, NS2 supporte du routage multi-chemins de type ECMP. En revanche, la répartition des paquets se fait dans un mode très simpliste appelé *round-robin*. Nous allons ici améliorer cet aspect en proposant une répartition par flux. Pour cela, il faudra modifier les paramètres de routage du fichier *multipathclassifier.cc* et également marquer vos flux via les agents TCP (vous serez donc amené à modifier le code C et OTCL pour interagir depuis l’interface TCL). Dans un premier temps, vous pourrez travailler sur une topologie (simple) de votre choix.

### Exercice 3 : Répartition par flux - à rendre pour le Vendredi 12 Janvier

**Q 1.** Observez sur un scénario de votre choix comment ECMP en mode round-robin peut dégrader les performances des flux TCP.

**Q 2.** Implémentez une répartition par flux et montrez que celle-ci règle les problèmes de la question précédente.

**Q 3.** Malgré cet effort, est-ce qu’ECMP produit nécessairement de meilleures performances qu’un routage mono-chemin ? Si ce n’est pas le cas trouvez un contre exemple.

**Q 4.** En reprenant la partie précédente, évaluez par comparaison et à “grande échelle” les bénéfices de votre implémentation d’ECMP dans un environnement réaliste.

**Q 5.** Essayez de mettre en place de la *QoS* pour favoriser certains flux dans un contexte ECMP. Définissez un scénario où le gain pour les flux privilégiés est significatif.

Synthétisez tous vos résultats les plus pertinents et leurs interprétations dans un court rapport de 5 pages (hors annexes de configuration et vos figures). Fournissez vos scripts de configuration et le code NS2 que vous aurez modifié.