

# Network Simulator 2 - Utilisation Basique

---

*Objectif* : Utiliser un simulateur réseau à événements discrets

*Notions* : Configuration TCL/NS2, premières analyses et interprétations de résultats

---

Commencez par installer NS2 ([http://nslam.sourceforge.net/wiki/index.php/User\\_Information](http://nslam.sourceforge.net/wiki/index.php/User_Information)) si celui-ci n'est pas déjà disponible en local sur votre machine ou via les dépôts ubuntu avec `apt-get install`. N'hésitez pas à vous appuyer sur l'abondante documentation en ligne disponible au sujet de ce simulateur ; notamment les deux références suivantes :

- <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- [https://dpt-info.u-strasbg.fr/~mathis/Enseignement/Reseau/TP\\_NS/](https://dpt-info.u-strasbg.fr/~mathis/Enseignement/Reseau/TP_NS/)

## Premières manipulations

### Exercice 1 : Première simulation

**Q 1.** Créez un graphe en étoile comprenant cinq noeuds feuilles. Vous utiliserez un routage à états des liens, et, dans un premier temps, les paramètres réseaux (débit, délais, taille file, etc) seront uniformes et laissés à vos soins.

**Q 2.** Mettez en place un flux applicatif de type UDP/CBR depuis chaque noeud feuille vers seulement une des cinq feuilles (notion d'agents émetteurs et récepteurs). Faites en sorte que le lien vers cette feuille réceptrice soit légèrement congestionné sans que ce soit le cas pour les autres liens.

**Q 3.** Visualisez votre simulation avec `nam` de telle sorte que le remplissage des files d'attente soit visible (notamment celle dont la capacité est insuffisante pour la charge).

**Q 4.** Essayez d'extraire avec des scripts et/ou des appels tcl des informations pertinentes au sujet de l'utilisation des liens, de leurs files d'attente en sortie et des pertes de paquets.

**Q 5.** Visualisez l'évolution de ces informations dans le temps au moyen du logiciel de tracé de courbe de votre choix : `xgraph`, `gnuplot`, `matplotlib`, etc.

**Q 6.** Reprenez cet exercice avec 93 noeuds ! Il faudra automatiser leur placement en étoile autour d'une clique de trois noeuds\*. Chaque noeud coeur aura une grappe de 30 noeuds directement connectés à lui et chaque noeud d'une grappe devra envoyer au moins un flux vers un noeud de chacune des deux autres grappes. Essayez d'équilibrer la charge sur les trois liens de coeur en définissant vos (au moins 180) flux de manière adéquate. Pour finir, arrangez vous pour que les trois liens de coeur soient très légèrement congestionnés et visualisez les.

---

\*. À ce stade, et dorénavant, vos scripts de simulations `tcl` ne seront plus générés manuellement mais via des outils de configuration automatique (de préférence avec des langages de haut niveau comme `bash`, `python`, `perl`, etc ou directement en `tcl` avec les structures de contrôle adéquates). En d'autres termes, vous automatiserez la génération des entrées de vos simulations (modèle(s) de réseau et de trafic) comme vous l'avez déjà fait avec le traitement des données en sortie.

## **Exercice 2 : Congestions et flux TCP**

**Q 1.**Définir un graphe à huit noeuds avec un lien de coeur “goulet d’étranglement” et six noeuds feuilles. Au niveau du trafic, vous établirez deux series de trois paires d’émetteurs/récepteurs TCP dans les deux directions (donc 12 flux en tout). Faites en sorte que les paramètres réseaux soient hétérogènes entre les trois chemins (RTT pour chaque couple émetteur/récepteur en particulier).

**Q 2.**Créez et visualisez une forte congestion sur le lien partagé impliquant les six flux TCP d’une des deux directions entre les noeuds feuilles. Faites en de même dans l’autre direction avec des paramètres TCP différents de la première.

**Q 3.**Analysez l’évolution de la charge du goulet d’étranglement dans le temps en fonction de plusieurs paramètres (notamment la version et/ou le paramétrage pour chaque flux). Vous pourrez analyser chaque direction du lien avec des objectifs différents, par exemple dans un sens il pourrait s’agir de la compétition entre flux TCP homogènes alors que dans l’autre direction ces mêmes flux seraient hétérogènes.

**Q 4.**Au moyen d’un “script de parsing”, analysez l’évolution de la fenêtre de congestion TCP de chaque flux précédemment établi et commentez l’équité du partage du goulet d’étranglement en fonction des paramètres de chaque flux (notamment la latence introduite sur chaque lien/chemin et les versions et/ou paramétrage TCP). De manière générale, commentez et justifiez les performances de chacun de vos flux TCP.

## **Manipulations avancées**

### **Exercice 3 : Projet**

**Q 1.**Rendez-vous sur moodle dans la rubrique projet (à venir prochainement si pas encore disponible). Les exercices précédents constituent la base du projet : cette base sera notée sur 8 pts au moins (a priori 3+5).