

# 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://nsnam.sourceforge.net/wiki/index.php/User\\_Information](http://nsnam.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 six 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 deux des six feuilles (notion d'agents émetteurs et receveurs). les deux destinataires devront recevoir la même charge.

**Q 3.** Visualisez votre simulation avec `nam` de telle sorte que le remplissage des files d'attente soit visible.

**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 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 104 noeuds ! Il faudra automatiser leur placement en étoile autour d'une clique de quatre noeuds\*.

### Exercice 2 : Congestions et flux TCP

**Q 1.** Définir un graphe à huit noeuds avec un lien de coeur "goulet d'étranglement" et deux fois trois paires d'émetteurs/récepteurs TCP dans une seule direction. Faites en sorte que les paramètres réseaux soient hétérogènes (RTT pour chaque couple de paire en particulier).

---

\*. 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. Comme avant, essayez d'équilibrer la charge (chaque grappe de noeuds doit être équilibré autour de la clique et émettre et recevoir la même quantité de trafic).

**Q 2.** Créez et visualisez une forte congestion sur ce lien impliquant plusieurs flux TCP entre les noeuds feuilles.

**Q 3.** Analysez l'évolution de la charge de ce lien dans le temps en fonction de plusieurs paramètres (notamment le type des flux TCP pour chaque couple de paire).

**Q 4.** Au moyen d'un "script de parsing", analysez l'évolution de la fenêtre de congestion TCP de chaque flux et commentez l'équité du partage du lien en fonction des paramètres réseaux de chaque flux/chemin (notamment la latence introduite sur chaque lien/chemin).

## Manipulations avancées

### Exercice 3 : Projet

**Q 1.** Rendez-vous sur moodle dans la rubrique projet. Les exercices précédents constituent la base du projet (noté sur 6 pts au moins).