

## Network Simulator 2 - Utilisation Basique

---

*Objectif* : Utiliser un simulateur réseau à événements discrets

*Notions* : Configuration TCL/NS2, premières analyses 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.di.unistra.fr/~merindol/Enseignement/Reseau/TP\\_NS/](https://dpt-info.di.unistra.fr/~merindol/Enseignement/Reseau/TP_NS/)

### Manipulations Basiques et Comparaisons TCP

#### Exercice 1 : Premières Simulations et Automatisation

**Q 1.** Créez un graphe en étoile comprenant six nœuds feuilles. Dans un premier temps, les paramètres réseaux (débit, délais, taille file, etc) seront uniformes et laissés à vos soins (voire ceux choisis par défaut par le simulateur).

**Q 2.** Mettez en place un flux applicatif de type UDP/CBR depuis chaque nœud feuille vers le nœud feuille suivant (modulo six). Avec un minimum de modification à ce premier scénario, faites en sorte qu'un seul et unique lien de votre réseau subisse des pertes.

**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 du lien incriminé).

**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 : `gnuplot`, `matplotlib`, `xgraph`, etc.

**Q 6.** Reprenez cet exercice avec cette fois 102 nœuds – dont 3 formant un triangle de cœur interconnectant les  $3 \times 33$  nœuds feuilles (répartis équitablement sur le triangle). À ce stade, et dorénavant, vos scripts de simulations `tcl` ne seront évidemment plus construits manuellement élément par élément mais via des outils de configuration automatique (de préférence avec des langages de haut niveau comme `python`, `bash`, `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 avez commencé à le faire avec le traitement des données en sortie. Il faudra notamment automatiser le placement des 99 nœuds feuilles en étoile autour du triangle central constitué de trois liens identiques aux capacités proportionnées aux liens vers les nœuds feuilles. Chaque nœud feuille (d'une des trois grappes du triangle) disposera d'un agent CBR émetteur qui devra envoyer un flux vers un agent récepteur attaché à un nœud de la grappe suivante. Chaque nœud sera impliqué dans exactement deux flux : un en réception et un en émission. Faites en sorte qu'un des trois liens du triangle de cœur soit plus chargé (mais pas saturé) que les deux autres et mettez en évidence cette différence sur un graphique.

## **Exercice 2 : Flux TCP et Contrôle de Congestion**

**Q 1.**Définir un graphe à huit nœuds avec un lien de cœur “goulet d’étranglement” et six nœuds feuilles (trois feuilles de part et d’autre du lien de cœur). Au niveau du trafic, vous établirez deux flux pour chaque couple d’émetteurs/récepteurs, les émetteurs seront positionnés d’un côté du lien de cœur et les récepteurs de l’autre. Chacun de vos six flux sera associé à une application de type FTP utilisant TCP au niveau transport. Faites en sorte que les RTT des chemins utilisés pour chaque couple émetteur/récepteur soient distincts pour vos premières expériences.

**Q 2.**Générez une saturation sur le lien de coeur et analysez le comportement de chaque flux (en particulier l’évolution de la fenêtre de congestion TCP ) et commentez l’équité du partage du goulet d’étranglement en fonction de la latence introduite sur chaque chemin. Inversement, sans congestion sur le lien de cœur, qu’observez vous lorsque ce n’est pas la fenêtre de réception qui limite le débit des flux ?

**Q 3.**Faites en sorte que les liens de votre simulation, en particulier le lien de coeur, disposent d’une haute capacité : montrez que les versions de base de TCP ont des difficultés à atteindre leurs débits maximaux théoriques.

**Q 4.**Pour finir, analysez l’évolution de la charge du goulet d’étranglement dans le temps en variant les paramètres de vos simulations. Utilisez par exemple différentes versions de TCP et comparez les entre elles et vis à vis de l’équité au sujet des RTT. Modifiez la nature des émetteurs/récepteurs TCP (**Tahoe**, (**New**) **Reno**, **Sack**, **Cubic**, **Vegas**, etc ainsi que la configuration de leurs paramètres internes) avec pour objectif d’améliorer les performances, en particulier le débit utile des flux (c’est à dire le rendement effectif de la liaison). Analysez ensuite la concurrence entre flux TCP différents lorsque les caractéristiques réseaux sont identiques sur les trois chemins. Enfin, modifiez la nature et la taille de la file d’attente du lien de cœur et commentez les nouvelles performances obtenues. Est-ce que vous observez des différences significatives ?

## **Manipulations Avancées : Analyse TCP**

### **Exercice 3 : Suite du Projet**

**Q 1.**Rendez-vous sur Moodle dans la rubrique projet (à venir prochainement si pas encore disponible). Les exercices précédents, notamment le second, constituent la base du mini projet à venir...