

NS2 - Simulations Plus Avancées

Objectif : Maîtriser un simulateur réseau

Notions : Analyses & interprétations de résultats dans un contexte réaliste

Manipulations basiques de NS2 : ≈ 18 pts - retours en binôme

Exercice 1 : Flux TCP, équité et performances - à rendre pour le jeudi 15 décembre

Q 1. Définir un réseau à huit nœuds avec un lien de cœur *goulet d'étranglement* et six nœuds feuilles (trois feuilles de chaque côté du lien de cœur). Au niveau du trafic, vous établirez six flux (pour trois fois deux couples d'émetteurs/récepteurs), les trois émetteurs seront positionnés d'un côté du lien de cœur et les trois récepteurs respectifs de l'autre. Vos six flux seront chacun associés à une application (par ex. de type FTP) utilisant TCP au niveau transport (la version basique de votre choix pour commencer, par ex. Tahoe). Faites notamment en sorte que les RTT des chemins utilisés pour chaque couple émetteur/récepteur soient suffisamment hétérogènes pour réaliser vos premières expériences sur l'équité de TCP vis à vis du RTT.

Q 2. Générez une congestion sur le lien de cœur (saturez le en jouant sur la fenêtre de réception paramétré côté émetteur) sur une période suffisamment longue pour mettre en évidence des résultats significatifs (que ce soit pour les RTT ou d'autres analyses plus originales). Analysez le comportement de chaque flux (l'évolution de la fenêtre de congestion TCP et le débit utile en particulier) et commentez l'équité du partage du goulet d'étranglement en fonction de la latence introduite sur chaque chemin. 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. Analysez l'évolution de la charge du goulet d'étranglement dans le temps en variant les paramètres de vos simulations (étudiez par exemple la charge de sa file d'attente). Utilisez notamment différentes versions de TCP et comparez les entre elles (à ce stade, il est préférable de revenir à des délais homogènes pour éviter les expériences croisées et multi-critères). Modifiez la nature des émetteurs/-récepteurs TCP (par ex. New Reno, Cubic, Vegas, Sack, Delayed Ack, etc. ainsi que la configuration de leurs paramètres internes) avec pour objectif d'améliorer les performances, en particulier le débit utile des flux (par rapport au cas idéal que vous utiliserez comme base de référence) et l'équité (vis à vis de la latence et entre les versions protocolaires). Analysez la concurrence entre flux TCP différents lorsque les caractéristiques réseaux sont identiques sur tous les chemins. Pour finir, modifiez la taille et éventuellement la nature de la file d'attente du lien de cœur (dans un intervalle suffisamment grand pour être concluant) et commentez les nouvelles performances obtenues. Est-ce que vous observez des différences significatives ? Essayez d'interpréter.

Q 4. Montrez que certaines versions de TCP sont plus extensibles que d'autres pour atteindre de grands débits (≥ 10 Gpbs). Pointez alors les éventuels défauts de ces variantes.

Q 5. Commentez et analysez en détail les résultats obtenus quand la charge de votre réseau évolue à la hausse (au moins sur le lien de cœur que vous surchargerez à dessein avec du trafic de fond et/ou davantage de flux TCP) : sa charge, sa file d'attente et le débit utile (i.e. efficace) des flux le traversant.

Faites une analyse spécifique pour les pires flux, ceux qui subissent le plus de re-transmissions (tel que leur débit utile soit bien plus faible que leur débit brut mesuré). Ils constitueront une base de flux TCP témoins illustrant les différences de comportement entre eux et éventuellement entre versions TCP. Pensez à mettre à profit les résultats obtenus dans les exercices précédents (que ce soit au niveau des files d'attente ou des variantes TCP) sans oublier d'évaluer vos simulations à différentes échelles de temps pour mieux comprendre la nature des congestions engendrées par le trafic injecté en plus.

Q 6.Bonus : Proposez une analyse originale de votre choix et essayez de mettre évidence un phénomène moins connu que ceux que vous aurez illustrés dans les questions précédentes (au sujet de TCP et/ou de leurs interactions avec des files d'attente).

Répondez en détail à toutes les questions ci-dessus. La base de l'évaluation pour cette seconde partie sera un rapport étoffé et illustré commentant et interprétant les résultats (environ 10/15 pages au maximum hors annexes de configuration). Vos configurations seront également commentées en détail pour justifier vos choix.

Fournissez vos scripts de configuration (pas nécessairement le produit TCL en résultant si vous avez utilisé un autre langage en amont) et justifiez tous vos choix réseaux (distributions statistiques, taille/-type des files, type de flux et configuration TCP, etc) en terme de réalisme. Justifiez aussi vos critères d'évaluation utilisés pour vos comparaisons. L'emploi de méthodes statistiques (a minima des intervalles de confiance) pour le traitement des données issues de vos simulations est vivement recommandé. Soignez vos représentations graphiques en y apportant un maximum d'explications et d'interprétation. Expliquez et analysez les résultats obtenus dans un rapport de 10-15 pages au maximum (hors annexes de configuration).

Votre rapport de synthèse décrira d'abord de manière précise et clair votre environnement de simulation (les paramètres réseaux utilisés et pertinents pour la couche transport) pour justifier le bien fondé et la pertinence de vos analyses (vos comparaisons doivent être "raisonnables" et incrémentales, par exemple mono-critère au moins dans un premier temps). 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 avec la charge des liens comme du côté applicatif avec le débit utile). Pensez en effet à considérer le débit utile plutôt que le débit brut dans vos analyses avancées.

Les bonnes questions à se poser sont les suivantes : est-ce que le partage de ressources est équitable (vis à vis de la latence ou entre versions distinctes) ? comment définir la bonne taille de file d'attente ? est-ce qu'il existe des versions de TCP plus performantes que d'autres dans un environnement donné ou plus généralement ? si c'est le cas, est-ce toujours équitable ou cela se fait au détriment d'autres versions moins agressives de TCP ?..et pour finir, n'hésitez pas à prendre un risque en sortant des sentiers battus.

Développement NS2 et extension ECMP : $\approx 2+$ pts bonus sur toute l'UE

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`, puis, dans un second temps, 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).

Exercice 2 : Répartition par flux - à rendre pour le vendredi 13 janvier (retours par groupe de quatre étudiants maximum)

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.

Synthétisez tous vos résultats les plus pertinents et leurs interprétations dans un court rapport d'environ 5-10 pages (hors annexes de configuration et vos figures). Fournissez vos scripts de configuration et le code NS2 que vous aurez modifié.