

# Protocoles de communications dans les réseaux Emergents : conception et évaluation Université de Pau

Cours de C. Pham



## Environnement de simulation OMNET++/INET

OMNET++ est installé sur les machines. Vous devez installer INET de la manière suivante :

Créer un répertoire INET dans votre espace personnel et positionnez vous dedans  
Copier le fichier `/install/INET.tgz` dans `~/INET`  
Décompressez le

La librairie INET qui fonctionne avec OMNET++ est maintenant installée. Vous pouvez dès maintenant regarder les exemples 3 à 5 sur la page :

<http://www.univ-pau.fr/~cpham/Enseignement.html>

Prise en main d'OMNET++, les différentes phases de la conception

- 1er exemple pas à pas: tic-toc
- 2e exemple: implantation d'un réseau de communication simple
- 3e exemple: l'extension INET (documentation sur INET) pour les réseau de communication, ARPTTest en détail
- 4e exemple: tutorial de TCP avec INET
- 5e exemple: implantation d'un scénario de communication TCP simple

## **TD/TP N°1 (un petit rapport sera à rendre)**

En partant de l'exemple « Round-Trip Time plot » traité dans le tutorial sur TCP (INET) nous allons commencer à analyser un peu plus en détails les résultats.

A/

Charger l'exemple `REDTest` avec des routeurs drop-tail (run 1). Lancer la simulation en mode Express jusqu'au temps 50s. Sortir de la simulation. Lancer `plove omnetpp.vec`.

- Tracer la courbe du `cwnd s1`.
- En quoi est exprimé `cwnd` ? Utiliser les « pre-plot filtering » pour l'exprimer de manière plus simple.
- Identifier les phases de *slow-start* et de *congestion avoidance*. Identifier la première perte. Quelle est la valeur de `cwnd` ? Expliquer ?
- Tracer sur une même courbe le `send seq s1` et le `rcvd ack s1`. Quels sont les paquets qui sont perdus ? Quand seront-ils retransmis ? Pourquoi les 2 courbes divergent-elles l'une de l'autre au fur et à mesure ? Valider votre réponse en proposant l'affichage une 3<sup>e</sup> courbe.
- Sortir de la simulation

Editer le fichier `omnetpp.ini` pour sélectionner le scénario `REDTestTh` (ce sera le réseau utilisé jusqu'à la fin du TD). Charger l'exemple `REDTest` de nouveau, lancer comme précédemment la simulation et `plove`.

- Tracer la courbe du débit.
- Où ce débit est-il mesuré ?
- Modifier dans le fichier `REDTestTh.ned` afin que le débit soit mesuré entre `s1` et `r1`.
- Relancer la simulation et afficher de nouveau le débit. Afficher sur la même courbe l'évolution du `cwnd` de `s1`. Ajuster les échelles si besoin (en utilisant le `pre-plot filtering`)

Sortir de la simulation. Renommer le fichier `omnetpp.vec` en `omnetpp.dt.vec`

B/

---

Charger l'exemple `REDTest` avec des routeurs `RED` queues `5..15pk` (run 2). Lancer la simulation en mode `Express` jusqu'au temps 50s.

Sortir de la simulation. Renommer le fichier `omnetpp.vec` en `omnetpp.1red.vec`

Charger l'exemple `REDTest` avec des routeurs `RED` queues `5..50pk` (run 3). Lancer la simulation en mode `Express` jusqu'au temps 50s.

Sortir de la simulation. Renommer le fichier `omnetpp.vec` en `omnetpp.2red.vec`

Lancer `plove` avec les 3 fichiers `.vec`

Constatez les différences en traçant les courbes qui vous semblent les plus représentatives. Montrez les à l'instructeur.

C/

---

Changer les débits des liens entre les machines terminales et les routeurs de 10Mbits/s à 100Mbits/s.

Re-exécuter la simulation avec les routeurs `drop-tail` (run 1), regarder et interpréter les résultats.

Dans le fichier `omnetpp.ini`, `s1` n'envoie qu'un seul flux TCP. Faites en sorte qu'il envoie un 2<sup>e</sup> flux TCP à `t=3s` (par rapport au flux de `s1`). De même, la machine `s2` n'envoie pas de flux. Faites en sorte qu'elle envoie un flux TCP à `t=20s`.

Ajouter une collecte de débit pour le lien entre `s2` et `r1`.

Re-exécuter la simulation avec les routeurs `drop-tail` (run 1), regarder et interpréter les résultats.

Mettre le délai entre les 2 routeurs de 20ms à 180ms.

Re-exécuter la simulation avec les routeurs `drop-tail` (run 1), regarder et interpréter les résultats.

D/

---

Dans le fichier `omnetpp.ini`, changer quelques paramètres du RED pour le run 1. Essayer de voir l'impact que cela a sur les pertes, le débit,...

**Attention : notez les paramètres initiaux pour pouvoir les remettre ensuite !**

E/

---

Dans cette partie nous allons essayer de mettre en évidence les problèmes liés au haut-débit. Dans le fichier `REDTestTh`, mettre tous les débits des liens à 200Mbits/s (sans toucher au délai, celui entre `r1` et `r2` était déjà fixé à 180ms).

Re-exécuter la simulation avec les routeurs drop-tail (run 1). Renommer le fichier `omnetpp.vec` en `omnetpp.dt.vec`. Lancer `plove` et afficher le débit. Afficher ensuite le débit moyen.

Relancer l'exemple `REDTest` avec des routeurs RED queues 5..15pk (run 2). Renommer le fichier `omnetpp.vec` en `omnetpp.redqueue.vec`. Lancer `plove` et comparer les débits moyen des 2 simulations, est-ce très différent ?

F/

---

Dans cette partie nous allons essayer de provoquer des pertes massives.

En vous inspirant de l'exemple `RouterPerf` (`RouterPerf.ned` et `omnetpp.ini` dans `../RouterPerf`), remplacer dans le fichier `REDTestTh.ned` `s4` par un récepteur de type `BurstHost`, et ajouter un émetteur raccordé à `r1` de type `BurstHost`. Cet émetteur va commencer à émettre à  $t=30s$ , va émettre un paquet de 800 octets toute les  $10^{-5}s$  et va envoyer en tout 500000 paquets.

Re-exécuter la simulation avec les routeurs drop-tail (run 1). Afficher le débit moyen de `s1`. Identifier où se produisent les pertes. Expliquer pourquoi ?

Modifier le fichier `.ned` pour que les pertes se situent principalement au niveau du routeur `r1` en entrée.

Re-exécuter la simulation avec les routeurs drop-tail (run 1). Afficher le `cwnd` des 2 flux de `s1` et de celui de `s2`. Afficher le débit moyen de `s1` et étudier la courbe.