

# Simulation et Algorithmique des Systèmes distribués

## Charleroi — Projet

Luc Onana Alima – Stefan Beauport

19 mars 2009

Version du 19 mars 2009

## 1 Enoncé

On souhaite simuler un réseau de télécommunication. Les grandes lignes de la construction du réseau sont données ci-après. Le comportement général du réseau est également explicité. Il vous est demandé de jouer sur les paramètres du système (taux de paquets perdus, latence des connexions) afin de faire apparaître des situations intéressantes.

Les explications suivantes s'appuient la topologie représentée par la figure 1. Plusieurs agents sont connectés entre eux selon le schéma ci-après. Un certain nombre d'hôtes sont connectés à un agent. Pour qu'un hôte envoie un message à un autre hôte, le message est routé par l'agent. Par exemple, si l'hôte 4010 souhaite envoyer un message à l'hôte 2022, le message passera par l'agent 4000, ensuite par l'agent 3000 et l'agent 2000 pour enfin arriver à l'hôte 2022. Quand un message est envoyé, le destinataire envoie un accusé de réception à la source. Si la source n'a pas reçu d'accusé avant un certain TimeOut, elle décide de réexpédier le message.

Supposons que le temps de traitement d'un message soit instantané, si le message est expédié du hôte 4010 au temps  $t_0$ , il arrivera en 2022 au temps  $t_0 + 5 + 20 + 10 + 5$  à savoir  $t_0 + 40$ . Si le temps de traitement d'un message n'est pas instantané, il est possible que des messages soient retardés dans les agents, et le temps nécessaire pour atteindre 2022 peut donc être plus élevé que 40 unités de temps. Selon le temps de traitement d'un message, l'hôte 4010 recevra l'accusé au moins 80 unités de temps après l'expédition du message original.

La simulation que vous réaliserez s'appuyera sur la topologie représentée à la figure 2. La topologie du réseau proposée est fixée, le nombre d'agents et les liaisons entre ceux-ci ne changent pas. Dans un premier temps on considérera que les valeurs des liens ne changent pas et que les chemins peuvent être calculés à l'avance. Dans un second temps, les valeurs de ces liens pourront changer, et les chemins devront être recalculés en fonction. Plus d'information à ce sujet sera donnée dans la section Distance Vector Algorithm.

Tout hôte peut décider de communiquer avec un hôte relié au même agent, ou à un agent différent.

### 1.1 Contraintes du système

Au début de la simulation, tous les hôtes sont présents. On génère leur scénario, et la simulation commence. Il est possible que des messages soient perdus brutalement (selon le paramètre).

### 1.2 Les paramètres du système

Il doit être possible de spécifier les paramètres suivants :

- Le taux de perte brutale effectuée par un agent (par exemple : 5%)
- La valeur du timeOut. Temps qu'on attend avant de rémettre un paquet. ( Par exemple, une valeur  $> 80$ )
- Le temps de traitement d'un message dans un agent ou un hôte. Traiter un message, c'est décider de l'envoyer vers un autre agent par exemple, prend un temps de  $x$  unités de temps. Valeurs proposées : 0 ; 0,5 ; 1. Une valeur de 0,5 signifie qu'en une unité de temps, un agent peut traiter 2 messages. S'il en avait plus à traiter, ceux-ci resteraient (attendraient) dans la file d'attente. Une valeur de 0 signifie un traitement instantané.
- Le nombre d'hôtes reliés à un agent. Par exemple, 100.
- Temps-Inter-Envois. Il s'agit du temps entre deux envois d'un hôte du système. Par exemple une valeur aléatoire  $\in ]0, 200]$ .
- Le taux de messages d'un hôte qui seront à destination d'un hôte relié à un autre agent. Par exemple, 75%.
- Durée de la simulation (en unités de temps)

FIG. 1 – Schéma utilisé pour l'explication du projet.

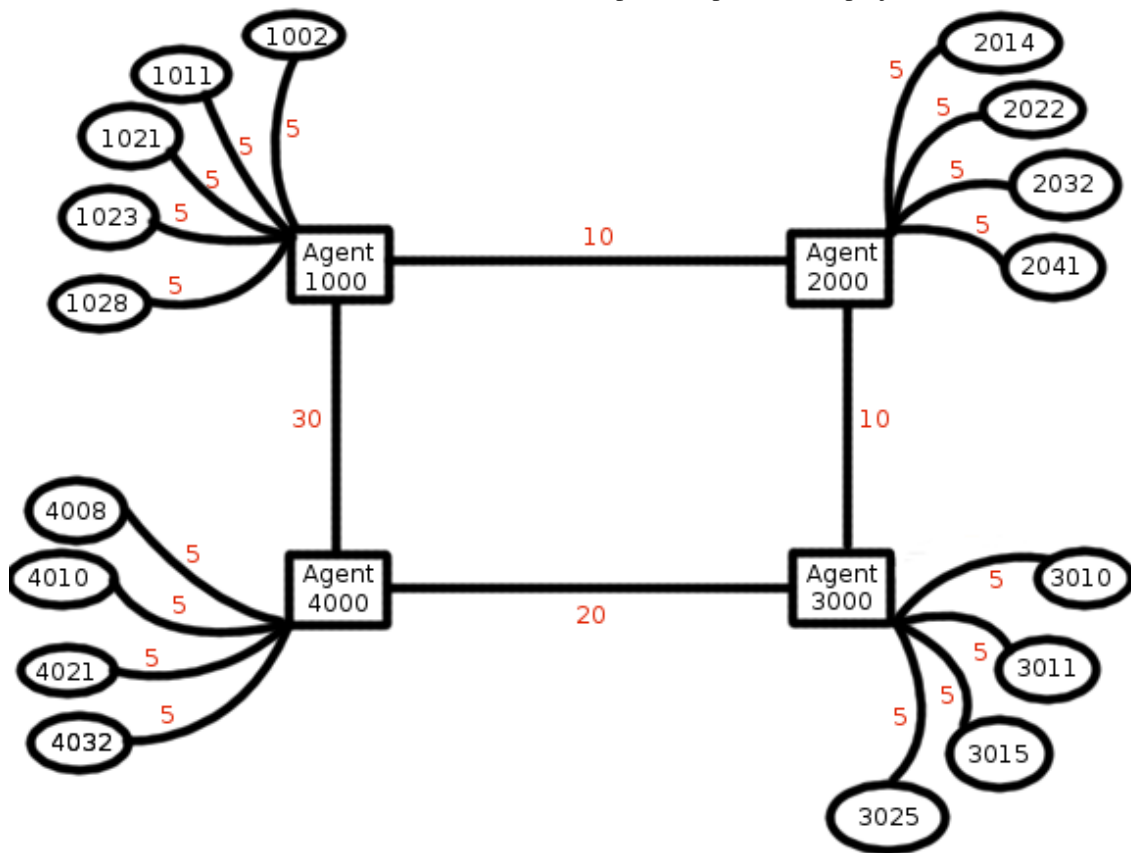
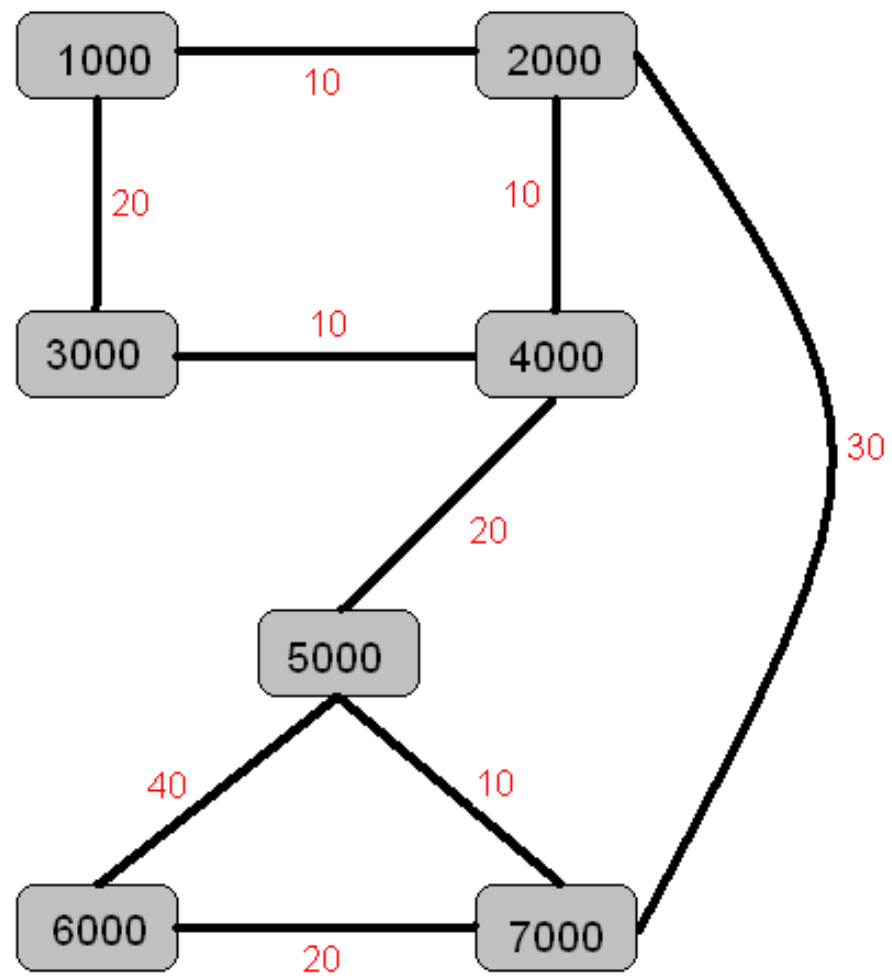


FIG. 2 – Topologie réseau à simuler. La valeur des liens représente la latence.



- La taille des buffers des agents. Par exemple, une taille infinie (pas de pertes), une taille de 100 ou de 1000 messages. Si un agent reçoit un message alors que son buffer est plein, il peut ne pas considérer le message. Dans un premier temps, on peut considérer une taille de buffer identique pour chaque agent, on peut ensuite spécifier des valeurs différentes pour chaque agent et voir l'implication.

### 1.3 Statistiques

Voici quelques statistiques intéressantes :

- Quel est le temps moyen entre l'émission originale d'un paquet, et la réception de l'accusé correspondant ?
- Quel est le pourcentage de paquets réémis ?
- Il y a deux raisons pour qu'un paquet soit perdu. Soit il s'agit d'une perte méchante (on a décidé que par agent, 5% des messages étaient perdus), soit il s'agit d'une perte due au fait que le buffer d'un agent était plein et n'a pas pu accueillir le message. Quel est le pourcentage de messages perdus pour chacune des deux raisons ?
- Combien y a-t-il eu de TimeOut trop courts ? C'est-à-dire des messages réenvoyés car on pensait que le message était perdu, alors que c'était juste parce que le message et son accusé étaient retardés.
- Quel est le temps moyen qu'un message passe dans une file d'attente ? (en absolu, et par rapport à son voyage complet)
- Quel est le taux moyen d'utilisation des buffers des agents ? (Par exemple, ils sont remplis à 80% en moyenne.) Quelle est la moyenne pour chacun des agents séparément ?

### 1.4 Distance Vector Algorithm

Comme expliqué plus tôt, dans un premier temps on peut considérer que la valeur des liens (exprimée ici en terme d'unité de simulation entre l'envoi d'un message d'un agent et la réception de ce message par l'agent destinataire) permet de calculer dès le départ le chemin qu'empruntera chaque message échangé entre chaque couple d'agents. Dans ce cas, le choix du chemin se fait uniquement en considérant la latence entre deux agents. On ne considère donc pas le taux d'occupation des agents le long du chemin, ou le taux de saturation des buffers. Effectivement, si un chemin est considéré comme meilleur car il propose une latence basse, il se peut que des agents sur ce chemin soient saturés car beaucoup de messages empruntent ce chemin. Il en résulte une perte de messages, ce qui rend ce chemin moins bon... Ceci nous permet de déduire que le chemin qui a la plus petite latence n'est pas forcément le meilleur, étant donné le trafic actuel dans le réseau.

Dans un second temps, nous vous demandons de considérer le principe de l'algorithme **Distance Vector** (voir fichier pdf). Lorsqu'un agent doit faire suivre un message vers un agent destinataire (final), il choisit à quel agent intermédiaire il va transférer le message conformément à l'algorithme proposé. La notion de « coût » utilisée dans l'algorithme peut être basée sur une combinaison des temps de latence et des taux de saturation des buffers des agents.

Les agents dialoguent entre eux pour échanger leurs informations de routage en utilisant des messages (tout comme les autres communications provenant des hôtes). Il n'est pas interdit cependant qu'un agent considère en priorité les messages de « routage » par rapport aux messages de communication des hôtes (FIFO pour tous les messages autres que messages de routage).

La période d'initialisation des tables de routage des agents peut se faire avant le début des envois de messages par les hôtes.

Les agents ne doivent échanger leurs tables de routage avec leurs voisins que lorsqu'il y a un changement significatif dans les coûts des liens (pas à chaque petit changement), mais au moins une fois toutes les  $x$  unités de simulation.

## 2 Le rapport

Le rapport doit contenir un certain nombre d'informations.

1. Vous devez spécifier chaque type d'événements qui peut survenir sur les hôtes, et sur les agents. Pour chacun de ses événements, veuillez décrire quelles sont les actions que vous avez décidé d'effectuer. Cette partie du rapport permet de comprendre comment est structuré votre système sans devoir analyser le code source en profondeur.
2. Vous devez utiliser votre outil.  
Proposez une certaine configuration du système (valeurs pour les différents paramètres) et montrez-nous grâce aux simulations, que c'est une situation à problème car, par exemple, il y a trop de paquets perdus. Proposez une solution pour améliorer la situation (en modifiant les paramètres), et montrez-nous que la situation, grâce aux simulations, est effectivement meilleure. Vous pouvez ainsi proposer deux situations à problème et proposer une

solution pour chacune.

Soit un trafic que vous aurez défini et un timeout (pas trop grand), quelle est la taille minimale des buffers des agents pour qu'il y ait moins de 20% de message réémis ? (A tester avec des pertes brutales, et sans pertes brutales.)

Soit ce trafic défini et le timeout fixé, donnez un graphe montrant l'implication de la taille des buffers sur le nombre de messages réémis. (Pour quelques simulations, notez la taille des buffers sur l'axe des abscisses et le nombre de messages réémis.) Donnez également un graphe montrant pour ces mêmes simulations l'implication de la taille des buffers sur le temps moyen total écoulé entre l'émission d'un message original et la réception de l'accusé correspondant.

Essayez de dégager une situation qui montre que l'utilisation du **Distance Vector Algorithm** est efficace. C'est-à-dire, dégagez une configuration particulière de la simulation où la version qui n'utilise pas le vecteur de distance propose des pertes de messages plus importantes que la version qui utilise le vecteur de distance.

3. Donnez quelques commentaires généraux pour interpréter vos graphes.

### 3 Dates importantes

- Le 3 avril 2009 : Avertir l'assistant par e-mail des groupes de travail (2 à 3 personnes par groupe).
- Le 7 mai 2009 : Remise du projet (code source, compilable java 6) et du rapport (version pdf) par e-mail, le tout dans une archive (Zip ou autre).