

TP3 – JAV : Simulateur d'un réseau de senseurs

I) Introduction

L'objectif de ce travail est avant tous de se familiariser avec JAVA et notamment avec la modélisation. Il s'agira en TD de proposer une architecture logicielle permettant de réaliser une simulation d'un réseau de senseurs. En TP, vous aurez à réaliser une implantation du simulateur à partir d'interfaces qui vous seront données. Le réseau simulé est assez « simpliste », l'objectif étant de donner un cadre à la conception d'une architecture orientée java.

II) Simulateur à événements discrets

Les réseaux, comme beaucoup d'objet du monde réel, peuvent être étudiés au moyen d'un *simulateur à événements discrets*. Il s'agit d'un programme dans lequel le temps est discrétisé, et une horloge virtuelle interne au programme gère l'ensemble des événements qui peuvent se produire dans le système réel modélisé dans le simulateur. Il n'y a aucun lien entre le temps discret et le temps réel d'exécution. A chaque *pas (step)* – c'est à dire l'avancement d'une unité de temps virtuel, l'ensemble des effets nécessaires à la simulation du fonctionnement des composants du système est appliqué. On peut ainsi simuler un tronçon de route, à chaque *pas* de temps virtuel, le simulateur calculant le déplacement élémentaire de chaque véhicule en fonction de sa vitesse, certain *pas* donnant lieu à des événements externes, comme le changement d'un feu de circulation, l'effet du climat etc... Dans ce travail, on modélisera un réseau de senseurs.

II) Réseau de senseurs en général

Un senseur est un élément actif réalisant des mesures physiques (par l'intermédiaire de capteurs) et capable de communiquer avec ces pairs. Un ensemble de senseurs communicants entre eux forment ainsi un réseau de senseurs. Les senseurs doivent être très peu coûteux et pour cela implantent généralement des protocoles de communication très simples et possèdent peu de capacité de calcul et de stockage.

L'objectif d'un réseau de senseurs, et de partager intégralement l'information produite par chaque senseur. Ainsi le réseau résiste globalement à la destruction partielle (ou la panne partielle du réseau) car l'information est massivement distribuée sur l'ensemble du réseau.

Le réseau de senseurs de notre exercice est très simple et n'a pas la prétention de ressembler à un véritable réseau (mais la philosophie y est).

III) Le réseau de senseurs à simuler

Dans notre exercice, nos senseurs sont interconnectés par des liens de communications point à point (par exemple: des connexions filaires) bidirectionnelles. Tous les senseurs et tous les liens du système présentent les mêmes caractéristiques.

III.1) Définition des entités du système

a) Chaque senseur possède :

- Un identifiant
- S sorties bidirectionnelles qui lui permettent d'être interconnecté à N autres senseurs. Un senseur peut ne pas avoir toutes ses sorties de connectées sans altération de son fonctionnement. Cette sortie est appelée un port.
- Un senseur génère un paquet à tous les pas de simulation

- Une file d'attente de taille N , de paquets en attente de traitement.
- Une unité de traitement des paquets, qui maintient comme information l'identification et la valeur des X derniers paquets reçus, ainsi que de tous les paquets qui sont passés.

b) Un paquet est une unité de transport d'information représenté par :

Un identifiant unique

La source qui a générée le paquet

Un paquet répliqué possède le même identifiant que sont répliqua.

c) Un lien qui permet d'interconnecter deux senseurs et présente comme caractéristiques :

L'identification des deux ports de la liaison.

Pour les deux sens de la liaison, une unité de stockage d'un paquet.

Le paquet émis au temps T , est reçu au temps $T+1$.

d) Le simulateur :

C'est le système qui permet de Permet d'instancier et d'exécuter tout le système. Il crée tous les éléments du système, il en identifie certain, il est capable de faire des analyse post-mortem.

III.2) Description de la dynamique du système

Le simulateur simule W pas d'itérations, dans le même pas, le simulateur actionne successivement :

- Toutes les captures de tous les senseurs actifs (simulation des événements discrets),
- Tous les liens,
- Tous les senseurs

a) Comportement du capteur

Lorsqu'un senseur doit générer un paquet, son capteur doit générer la mesure. Elle sera placée dans la file d'attente à l'activation du senseur.

b) Comportement d'un lien

Un lien est passif, il ne fait que stocker en mémoire deux paquets allant d'un port vers un autre. Il est entièrement actionné par les ports qu'il relie.

c) Comportement d'un senseur

Arrivée d'un événements

* Ces deux classes d'événements peuvent survenir.

- Génération d'un paquet d'information du capteur
- Arrivée de paquets sur les liens du réseau.

Le senseur traite tous les événements d'arrivée. Il traite en priorité le paquet d'information qu'il génère, puis l'arrivé des paquets arrivant sur le port, dans l'ordre des identifiant de liens.

Gestion d'une file d'attente

* L'événement traité est placé dans une file d'attente de taille P. Si la file est pleine, le paquet est détruit.

Traitement du paquet

- * Un paquet est extrait de la file d'attente
- * S'il est inconnu du senseur il y est enregistré
- * S'il est connu du senseur, il est détruit

Emission du paquet

- Si le paquet en traitement n'est pas détruit,
- * Son TTL (time to live) est décrémenté de 1
 - * Si le TTL est égal à 0, il est détruit
 - * Sinon il est émis sur tous les liens

d) Comportement du simulateur

Il s'exécute en trois phases :

- A l'instanciation, il fabrique tous les liens et les senseurs
- A l'exécution, il simule le fonctionnement du système
- A la terminaison, il analyse les résultats.

III.4) Résultats de simulation

L'objectif du système est de pouvoir simuler le comportement du réseau de senseurs sur différentes topologies.

Nous voulons comparer le comportement du réseau sur quatre topologies de 10 senseurs.

Anneau, centralisé, entièrement maillé, asymétrique (anneau de 5 nœuds et extrémité sur chaque nœud de l'anneau)

IV) Travail demandé

A l'aide des interfaces qui vous sont fournis, réaliser des classes qui les implantent. Proposer ensuite une classe « simulation » possédant une méthode main et tout ce qu'il faut pour lancer une simulation.

Réaliser une interface graphique permettant de lancer le processus de la simulation et de visualiser ses résultats...