Projet LO41 automne 2010

Le carrefour intelligent

utbm
université de technologie
Belfort-Montbéliard

Alexis Urvoy
Ahmed-Khalil Jerbi

Sommaire

Introduction

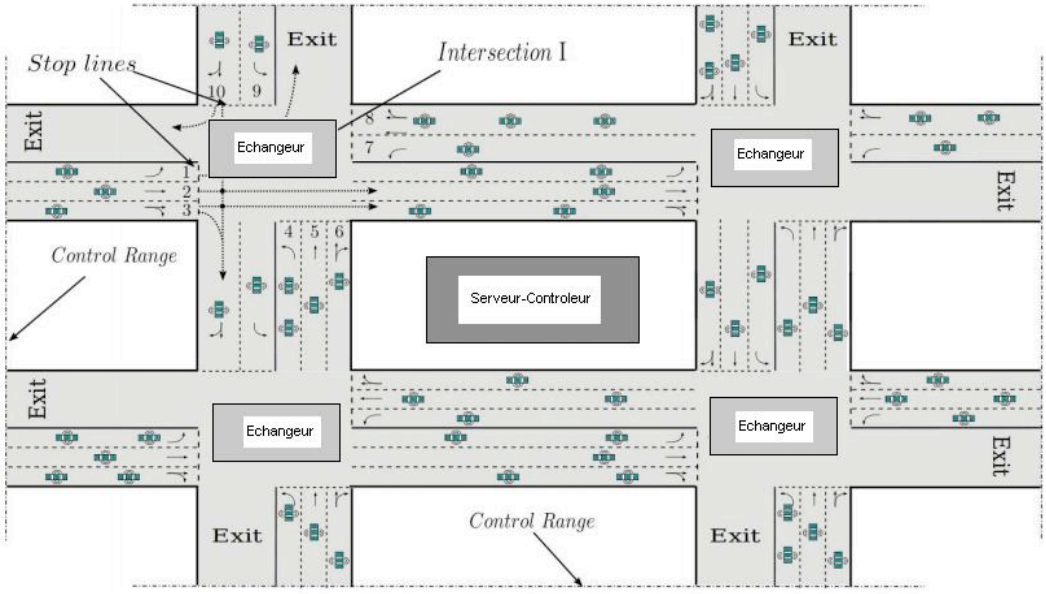
Dans le cadre de l'uv LO41, nous avions à développer un carrefour intelligent. Ce projet met en évidence toutes les techniques vues au cours du semestre, comme par exemple la synchronisation interprocessus.

I. Analyse du problème

Voici le sujet tel qu'il a été énoncé :

« Le projet concerne la simulation d'un ensemble de 4 carrefours. **Les carrefours sont dépourvus de feux de signalisation.** On suppose que tous les véhicules sont équipés d'un dispositif embarqué leurs permettant de communiquer avec l'infrastructure routière. Vis-à-vis de l'infrastructure, le carrefour est identifié par un échangeur. Cet échangeur communique avec les véhicules sur la base des informations qu'il collecte auprès du Serveur-Contrôleur. Ce dernier régule les flux en fonction de la position des véhicules et de l'état du trafic.

Exemple : Un véhicule arrivant dans un carrefour transmet à l'échangeur le plus proche son état (type de véhicule, vitesse, position, itinéraire,…). L'échangeur identifie le véhicule et transmets ensuite une requête au serveur-contrôleur. Ce dernier analyse l'état du trafic sur l'ensemble des 4 carrefours (nombre total des véhicules à chaque carrefour, type des véhicules,..) et répond à l'échangeur en fonction des contraintes de flux. Au final, celui-ci informe le véhicule demandeur. Certains véhicules sont prioritaires (Ambulances, pompiers, police,..) et doivent donc emprunter le chemin le plus court quelque soit l'engorgement du trafic. »



Notre problème consiste à gérer les différents carrefours pour permettre la meilleure circulation possible.

Plusieurs ressources critiques apparaissent. La première est le serveur-contrôleur. En effet, il peut être sollicité par les quatre échangeurs à la fois. Il faut gérer l'aspect de communication entre le serveur et les échangeurs de manière à ce qu'il y ait une communication a tour de rôle en évitant les problèmes de famine.

Les autres ressources critiques sont les quatre échangeurs. En effet, chaque échangeur doit gérer le flux de voiture approprié à son carrefour avec les même contraintes que pour le serveur-contrôleur.

Une autre chose à gérer sera les voitures prioritaires. Celle-ci doivent passer avant toutes les autres quoi qu'il arrive, et emprunter le chemin le plus court. Il faudra donc pouvoir faire la différence entre un véhicule prioritaire et un véhicule ordinaire.

Un autre problème qui se pose est celui de la taille limite des routes à l'intérieur du carrefour. D'une part, le serveur-contrôleur doit aiguiller les voitures en fonction du chemin le plus court (en terme de temps et/ou de distance). Et d'autre part, si les routes sont pleines, les voitures doivent attendre que des places se libère pour rentrer sur le carrefour.

II. Solutions envisagées

A. Les structures de données

Générateur de voitures

Le rôle du générateur de voitures est de créer des voitures et de les envoyer dans les entrées du carrefour. Chaque voiture est considérée comme un message, contenant cinq champs qui sont : sa voie de actuelle, sa voie d'arrivée, l'échangeur avec lequel elle doit communiquer, son numéro et le type de voie sur lequel elle est (voie de départ, ou intérieure).

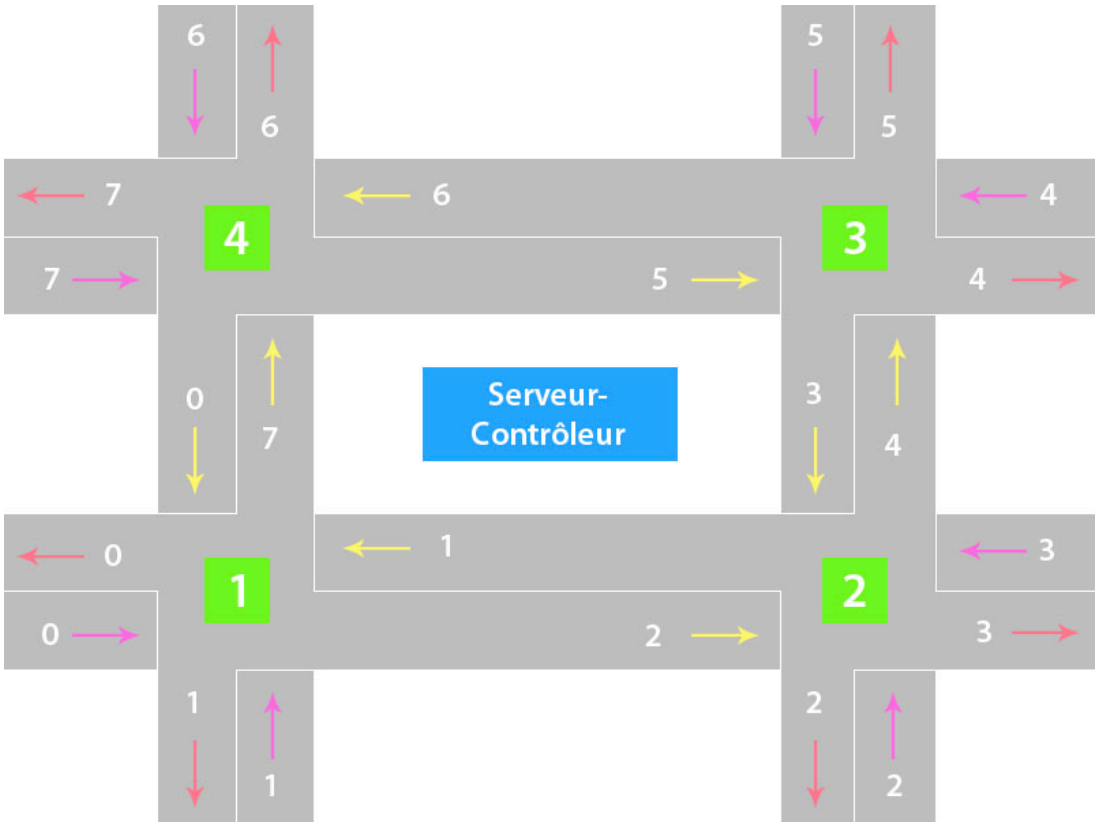
Echangeurs

Chaque échangeur est un processus qui doit gérer le carrefour qui lui ait attribué. En effet, il doit gérer les voitures présentent dans les quatre voies dont il s'occupe. Ces voies sont en fait des files de messages, sur lesquelles circulent des messages (les voitures). Le parcours des quatre files de messages s'effectue dans le sens inverse des aiguilles d'une montre. Lorsqu'il reçoit une voiture, l'échangeur communique avec le serveur par le biais d'une autre file de messages (commune aux quatre échangeurs), se met en attente de la réponse du serveur-contrôleur, puis envoie la voiture vers la file de message indiquée par le serveur.

Serveur-contrôleur

Le serveur-contrôleur quant à lui s'occupe de déterminer le chemin le plus court pour une voiture. Il reçoit un message de la part de l'échangeur, traite la requête et lui réponds.

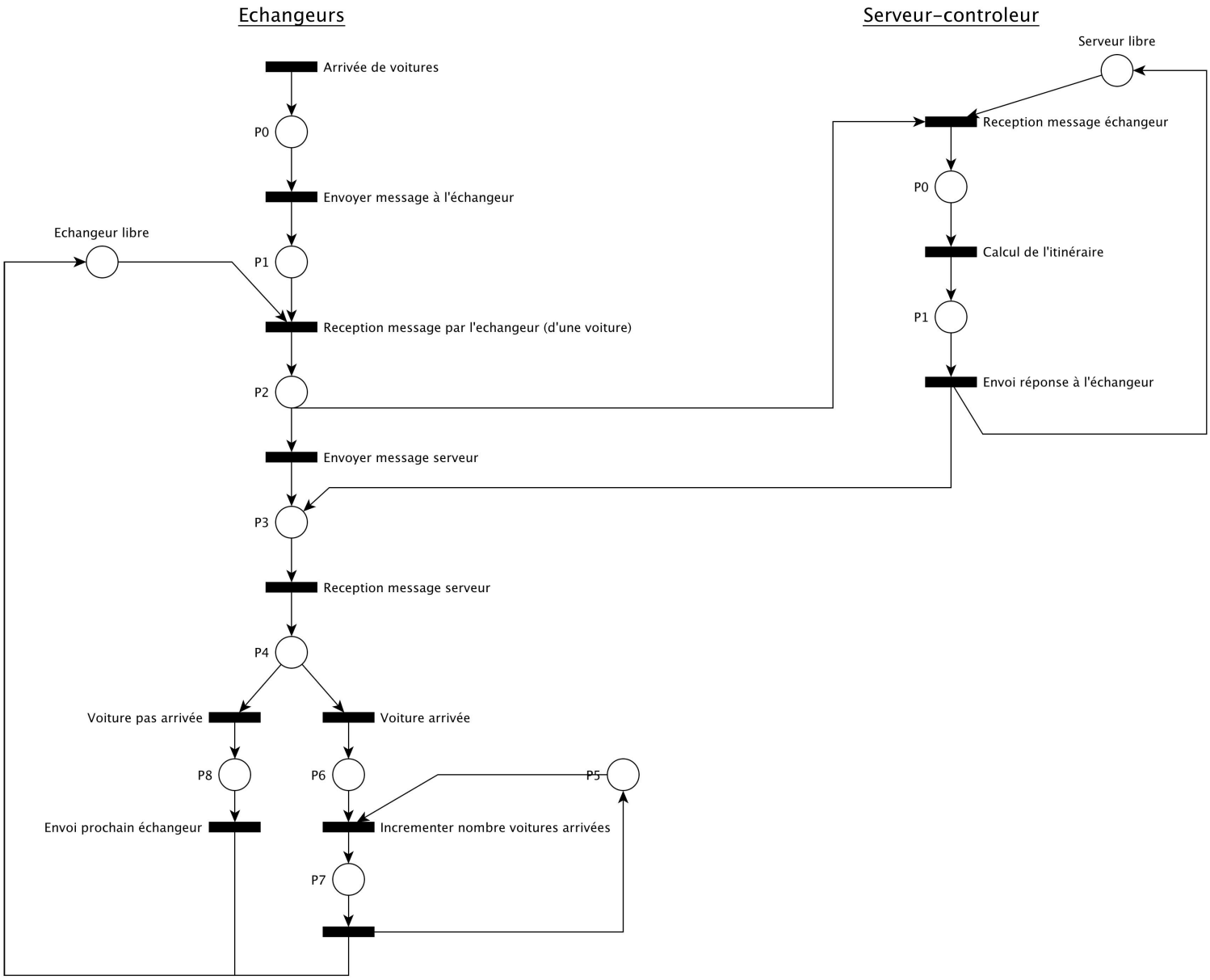
B. Représentation schématique de notre solution



Comme vous pouvez le constater, nous avons traité ce sujet de manière simplifiée en ne gérant qu'une seule voie par file, au lieu de trois.

Tout d'abord, on remarque que les files d'entrées (en violet) et de sortie (en rouge) ont le même numéro et ce afin de faciliter les opération par la suite.De même, les files intérieures (en jaune) et les files de sorties sur un même échangeur portent le même numéro. Ainsi il est plus facile de s'y retrouver et le modèle est plus cohérent.

C. Représentation du réseau de pétri



Conclusion

En conclusion nous pouvons donc dire que ce projet nous a permis de mettre en application toutes les technique vu en cours et d'appréhender une partie des contraintes de la programmation système. Le fait d'utiliser Open Solaris et Solaris nous a permis de découvrir un nouveau système d'exploitation.