



RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET
POPULAIRE



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ 20 Août 1955 – Skikda

Faculté des Sciences

Département d'INFORMATIQUE

Mémoire présenté en vue de l'obtention d'un Diplôme
Master 2

Spécialité : Systèmes informatiques

THÈME :

<p>Modélisation d'un feu de circulation par les Nets within Nets</p>

Réalisé par :

- *Steila Roukaya*

Encadré par:

- *Dr KISSOUM Yacine*

Année universitaire : 2021-2022

Remerciements

Avant de présenter ce travail, je tenter à remercier Dieu le tout puissant, de je avoir permis d'arriver à ce niveau d'étude et aussi pour avoir donné la force, le courage et la patience pour accomplir ce travail. 'Dieu merci'.

Á mes chers parents, mon fierté et bien sur la source de mon réussite, sans eux rien n'aurait pu être facile, je prie mon dieu dès les gardes et les protège.

Je tiens à remercier mon encadreur, *Dr KISSOUM YACINE*, pour sa patience, Et surtout pour sa confiance, ses remarques et ses conseils, sa disponibilité et sa bienveillance.

Merci à saker loubna et boulemche mohamed lamine d'essayer de m'apprendre de nouvelles choses dans le stage.

Merci à ma sœur et à tous ceux qui ont contribués de près ou de loin dans l'élaboration de ce travail.

Dédicaces

Je dédie ce mémoire :

À mon père et ma mère qui m'ont beaucoup aidée et encouragée durant mes études, qu'Allah les protègent.

À ma soeur khawla pour sa présence à mon côté.

À mes frères abd el hak, housine et abd el nafaa.

À ma soeur khadidja.

À toute ma famille.

À mes chers 7 personnes préférées Bangtan.

À mes amies Asma, youssra, nardjes, Aya, racha, rania et khadidja.

À tous les ARMYs qui m'ont encouragée dans les moments difficiles.

À tous ceux qui m'ont aidé de près ou de loin à accomplir ce Travail.

Et à moi-même évidemment.

Résumé

Dans les prochaines années à venir, les réseaux véhiculaires seront capables de réduire significativement le nombre d'accidents via les messages d'alerte échangés entre les véhicules de proximité. Dans ce travail nous avons étudié les réseaux de véhicules VANETs, qui connaissent de plus en plus d'intérêt aussi bien en recherche qu'en développement.

Ce mémoire présente un cadre conceptuel basé sur l'architecture multi-agents pour faciliter la simulation des VANETs dans les environnements intelligents dynamiques. L'architecture est modélisée à l'aide des nets within nets et simulé comme des scénarios réels en utilisant l'outil Renew.

Nous ferons une simulation d'un système de contrôle des feux de circulation intelligents. L'objectif est d'apporter une solution pour réduire les problèmes de congestion et les retards d'attente, et d'améliorer le système de feux de circulation traditionnel.

Mots clés

Les réseaux dans les réseaux, Système multi agents, RENEW, réseaux véhiculaires, réseaux de petri, Gestion du trafic, Système de feux de circulation intelligents, Intersection.

Abstract

In the next few years, vehicular networks will be able to significantly reduce the number of accidents via alert messages exchanged between nearby vehicles. In this work we have studied the vehicular networks VANETs, which are gaining more and more interest in both research and development.

This work presents a conceptual framework based on multi-agent architecture to facilitate the simulation of VANETs in dynamic intelligent environments. The architecture is modeled using nets within nets and simulated as real scenarios using the Renew tool.

We will do a simulation of a smart traffic light control system and the objective is to provide a solution to reduce congestion problems, waiting delays, and to improve the traditional traffic light system.

Keywords

Nets Within Nets, Multi Agent System, RENEW, vehicular ad hoc, Petri Nets, Traffic Management, Intelligent Traffic Light System, Intersection.

ملخص

في السنوات القليلة المقبلة ستصبح شبكات المركبات قادرة على تقليل عدد الحوادث بشكل كبير من خلال تبادل رسائل التنبيه بين المركبات القريبة من بعضها في هذا العمل درسنا شبكات المركبات بصفة عامة و التي تظل تكتسب المزيد من الإهتمام في كل من البحث و التطوير تقدم هذه المذكرة إطاراً مفاهيمياً يعتمد على مخطط تعدد الوكلاء لتسهيل تصميم شبكات المركبات في البيئات الذكية الديناميكية، كما تم تصميم المخطط باستخدام نموذج الشبكات داخل الشبكات و محاكاته كسيناريوهات حقيقية باستخدام أداة

RENEW

سنقوم بمحاكاة نظام ذكي للتحكم في إشارات المرور و الهدف هو توفير حل لتقليل مشاكل الإزدحام و طابور الإنتظار، كما و تحسين نظام إشارات المرور التقليدية المعتادة.

الكلمات الدالة

الشبكات داخل الشبكات، نظام تعدد الوكلاء، شبكات المركبات، شبكات بتري، التحكم في المرور، نظام إشارات المرور الذكي، تقاطع الطرق

SOMMAIRE

Chapitre I Introduction Générale.....	10
Introduction Générale.....	11
Chapitre II Généralités sur les réseaux VANETs.....	14
1. Introduction	15
1. Définition des réseaux	15
a. Réseau Ad-hoc	15
b. Réseau VANET.....	16
2. Architecture du réseau VANET.....	16
a. Entités de communication	16
i. Road Side Unit.....	16
ii. Autorité centrale	16
iii. On Board Unit	17
b. Modes de communication.....	17
i. Communication en mode Ad-hoc (V2V).....	17
ii. Communication en mode infrastructure (V2I).....	18
ii. Communication en mode hybride	18
c. Types de messages	19
i. Messages de contrôle	19
ii. Messages d'alerte.....	19
3. Les caractéristiques des VANETS	19
4. Les services offerts par les réseaux VANET	20
a. Les services liés à la sécurité routière.....	20
b. Les services liés au confort.....	20
5. Les applications des VANETS.....	21
6. Classification des protocoles de routage dans les réseaux VANET	23
a. Le routage basé sur la topologie	23
i. Les protocoles proactifs	23
ii. Les protocoles réactifs	24
iii. Les protocoles hybrides	25
b. Les protocoles de routage basés sur la géographique	26
7. Conclusion	28
Chapitre III Analyse et conception.....	29
SECTION 1 : OUTILS DE CONCEPTION	30
1. Introduction	30

2. Les Réseaux de Petri	30
<i>a. Concepts de base</i>	31
<i>b. Définition formelle</i>	32
<i>c. Définition informelle</i>	32
<i>d. Marquage</i>	33
<i>e. Franchissement d'une Transition :</i>	33
<i>f. Propriétés des Rdp</i>	33
<i>g. Les méthodes d'analyse des réseaux de Petri</i>	34
<i>h. Quelques extensions des réseaux de Petri</i>	35
3. Le paradigme (nets-within-nets)	37
<i>a. Les réseaux de référence</i>	37
<i>b. Modélisation de la mobilité</i>	38
<i>c. Multi Agent Architecture MULAN</i>	40
4. Outils de modélisation des réseaux de Petri	41
Section 2 : Conception (Cas D'étude)	42
1. Introduction	42
2. Travaux connexes	42
3. Description détaillée	43
4. Conclusion	48
Chapitre IV Implémentation.....	49
1. Introduction	50
2. Outils de conception	50
3. Langage de programmation utilisée	52
4. Plateforme utilisée	52
5. Environnement de développement	54
6. Interface	55
7. Conclusion	57
Conclusion générale	58
Conclusion générale	59
Bibliographie	61

Liste des figures

<i>Figure 2.1 : Exemple d'un réseau VANET.</i>	16
Figure 2.2: les entités de communication.	17
<i>Figure 2.3 : les modes de communication.</i>	18
<i>Figure 2.4 : réduire les accidents.</i>	21
<i>Figure 2.5 : Notifications des accidents.</i>	22
<i>Figure 3. 1: Un réseau objet contenu dans un réseau système</i>	39
<i>Figure 3. 2: Les réseaux système et objet après le franchissement.</i>	39
<i>Figure 3. 3 : Système multi-agents modélisé avec les réseaux dans les réseaux.</i>	41
<i>Figure 3.4 : Environnement de système.</i>	44
<i>Figure 3.5 : Structure de l'agent d'intersection.</i>	45
<i>Figure 3.6 : Structure de l'agent de véhicules.</i>	45
<i>Figure 3.7 : Le protocole de l'agent d'intersection.</i>	46
<i>Figure 3.8 : protocole de véhicule.</i>	47
<i>Figure 3.9 : exemple de simulation.</i>	48
Figure 4.1 : Renew GUI, Petri net.	50
Figure 4.2: Interface graphique de jade	54
Figure 4.3: IntelliJ IDEA IDE	55
Figure 4.4: Interface de logiciel.	56
Figure 4.5: fenêtre de l'agent "Sniffer".	56
Figure 4.6: message d'exécution.	57

Liste d'abréviations

ACC	Agent Communication Channel	JVM	Java Virtual Machine
ACL	Access Control List	LSR	Link State Routing
AMS	Agent Management System	MANET	Mobile Ad-hoc Network
AODV	Ad-hoc On Demand Distanc Vector	MPR	MultiPoint Relay
ASTAR	Anchor-based Street and Traffic Aware Routing	MULAN	Multi-Agent Nets architecture
CA	Central Authority	OBU	On Board Unit.
CTM	Cell Transmission Model	OLSR	Optimized Link State Routing Protocol
DF	Director Facilitor	RdP	Réseux De Pétri
DSR	Dynamic Source Routing	RdPT	Réseaux de Pétri Temporisés
DSR	Dedicated Short Range Communications	RENEW	Reference Net Workshop
DVCAST	Distributed Vehicle Broadcast Protocol	RSU	Road Side Unit.
FIPA	Foundation for Intelligent Physical Agents	SMA	Système Multi Agents
FSR	Fisheye State Routing	STI	Systèmes de Transport Intelligents
GLS	Grid Localization Service	TA	Trusted Authority
GPS	Global Positioning System	TPN	Timed Petri Nets
GSR	Geographic Source Routing	UML	Unified Modeling language
Gy-TAR	Greedy Traffic-Aware Routing Protocol	V2V	Vehicular-to-Vehicular
HARP	Hybrid Ad-hoc Routing Protocol	V2I	Vehicular-to-Infrastructure.
HCB	Hard Core Bitch	VANET	Vehicular Ad-hoc Network
IoT	Internet Of Things	ZRP	Zone Routing Protocol
JADE	Java Agent Development Framework	JVM	Java Virtual Machine

Chapitre I Introduction Générale

Introduction Générale

À mesure que la technologie informatique devient plus avancée, l'un des principaux facteurs moteurs de son avancement est la réduction de la taille des systèmes informatiques. En réduisant la taille des systèmes informatiques, nous augmentons à la fois la puissance des plus grands systèmes en leur permettant d'utiliser plus de ressources tout en rendant l'utilisation de systèmes de plus en plus petits viables. Cela se voit historiquement dans le développement de l'ordinateur personnel, qui a alimenté de nombreuses avancées dans les technologies de mise en réseau. Aujourd'hui, de minuscules dispositifs de plus en plus puissants sont en cours de développement et peuvent être intégrés dans des systèmes plus vastes tout en possédant des capacités de communication réseau. Cet ensemble de nouveaux appareils a également conduit à de nombreuses innovations dans les technologies de réseau, qui sont désignées dans la catégorie générale de l'Internet des objets (IoT).

Une sous-catégorie de technologies IoT [1] comprend les systèmes de réseau mobile ad hoc (MANET). Comme leur nom l'indique, les systèmes MANET cherchent à connecter des appareils mobiles, généralement ceux déplacés par l'homme, les uns aux autres pour créer un réseau. Étant donné que seuls les appareils sont connectés et qu'il n'y a pas d'appareils d'infrastructure de gestion tels que des routeurs ou des tours de téléphonie, ces réseaux sont considérés comme des réseaux ad hoc. Les systèmes MANET permettent à n'importe quel appareil du réseau d'échanger des informations avec tous les appareils qui se trouvent actuellement autour d'eux à tout moment.

Les MANET ont conduit au développement d'un autre domaine des systèmes de réseaux mobiles qui cherchaient à connecter ensemble des appareils déplacés par des véhicules. Ce domaine était à l'origine traité comme une sous-catégorie de MANET [2], mais au fur et à mesure de son enquête, il a été découvert qu'ils faisaient face à de nombreux défis uniques et pouvaient apporter de nombreuses opportunités uniques. En conséquence, le domaine des réseaux ad hoc véhiculaires (VANET) a été développé pour aider à se concentrer sur certains des aspects uniques de cet environnement.

Récemment et à la lumière du développement actuel auquel nous assistons ces dernières années, l'utilisation des véhicules est devenue une nécessité pour la majorité de la population, et dans une tentative de créer une nouvelle ère pleine de confort et de luxe, les moyens d'y parvenir sont toujours en développement.

Les réseaux véhiculaires sans fil (VANET) sont un élément clé des systèmes de transport intelligents. Ils permettent aux utilisateurs d'avoir des informations sur leur itinéraire, sur la météo, l'état de la route, l'accès à Internet, ... tout en temps réel. Ces réseaux donnent également accès à différentes applications de détente et de divertissement comme transporter des données multimédia, avoir des informations touristiques, etc.

Le réseau VANET donne accès aux informations en temps réel et en privé. Ce fait a un grand impact sur le comportement de l'utilisateur, sur sa vie et sur celle des passagers. Ce réseau assure également la sécurité des utilisateurs en préservant l'anonymat et en sécurisant leurs données personnelles.

Avec tous ces développements auxquels nous assistons et le grand nombre des véhicules qui sont devenues dans tous les pays, cela a entraîné un important embouteillage dans toutes les régions, ce qui provoque le nombre des accidents et les risques de collisions entre véhicules surtout dans les croisements des flux de trafic, les intersections, les ronds-points et aussi une augmentation du nombre de morts, et même avec l'apparition des véhicules complètement autonomes qui ont présenté une révolution dans le domaine de la robotique mobile et le transport routier et qui peuvent créer des solutions pour ce genre de problèmes, les intersections représentent toujours des goulots d'étranglement pour les flux de trafic et pour la circulation des véhicules et le peur d'avoir des dégâts plus importants augmente, ce qui empêche les industriels de ce domaine de se lancer dans des projets réels, même qu'on peut trouver sur le marché des véhicules semi-autonomes.

Les feux de circulation développés depuis 1912 qui sont des dispositifs de signalisation conçus pour contrôler les flux de trafic aux intersections routières, aux passages pour piétons, aux trains et à d'autres endroits, peuvent sembler être une solution à ces problèmes, mais ils ne suffisent pas surtout dans les intersections car ils peuvent rendre vraiment la circulation plus difficile et les trajets peuvent prendre plus de temps que prévu. [3]

Au cours de ce travail nous essayerons de mettre en place une solution à quelques problèmes auxquels la circulation routière lourde utilisant les feux de circulation.

Dans ce travail nous essayerons d'améliorer l'approche conventionnelle en y apportant un nouvel aspect. Un aspect se basant sur un système multi agent.

Nous tacherons durant les différents chapitres de ce mémoire de développer notre point de vue en ce qui concerne la conception et la réalisation des feux de circulation intelligent. Pour atteindre cet objectif nous avons choisi de répartir notre travail en quatre différents chapitres :

Le premier chapitre est une introduction générale à travers laquelle on a expliqué le domaine inclus dans notre travail.

Le deuxième chapitre présente la définition des réseaux sans fil existant (Ad-hoc et VANET), l'architecture du réseau VANET, ses différentes caractéristiques et normes de communication, les services offres par les VANETs, ainsi que ses applications, Enfin en présenterons les protocoles de routage dans les VANETs.

Dans le troisième chapitre les réseaux de Petri sont présentés comme méthode de modélisation, le chapitre commence par un ensemble de définitions des différentes notions liés aux réseaux de Petri, ensuite les types de réseaux de Petri en générale et les réseaux de référence en particulier sont présentés, et en finis par une étude de cas détaillée sur notre problème et la solution qui nous proposons.

Durant le quatrième chapitre nous parlerons de la simulation de notre système, les logiciels utilisées et la solution qui on a obtenue.

Enfin, nous terminerons par une conclusion générale et des perspectives pour de nouveaux travaux.

Chapitre II Généralités sur les réseaux VANETs

1. Introduction

Les communications qui peuvent être établies par un véhicule joueront un rôle important dans les années à venir. Où les véhicules de demain ne se contenteront plus de détecter les dangers grâce à des radars ou des caméras, ils seront capables de recevoir des messages d'alertes envoyés par les autres véhicules ou par l'infrastructure (panneaux, portiques, etc.) et de transmettre ces informations à d'autres véhicules.

Avec l'augmentation des accidents de la route et des décès associés, ainsi que la pollution de l'environnement due à des voitures, un nouveau type de réseaux est apparu, qui sont connus sous le nom "Réseaux Véhiculaires (VANETs)". Ces réseaux est un type particulier des réseaux ad-hoc mobiles (MANETs). Ils constituent le noyau des Systèmes de Transport Intelligents (STI) [4] qui ont objectif principal l'amélioration de la sécurité routière en tirant profit de l'émergence de la technologie de communication et la baisse du coût des dispositifs sans fil. En effet, grâce à des capteurs installés à bord des véhicules, ou bien situés au bord des routes et des centres de contrôle, les communications véhiculaires permettront aux conducteurs d'être avertis suffisamment tôt de dangers éventuels. De plus, ces réseaux ne se contenteront plus d'améliorer la sécurité routière seulement, mais ils permettront aussi d'offrir de nouveaux services aux usagers des routes rendant la conduite plus agréable.

Dans ce chapitre, nous présentons les réseaux VANETs, les différents types de services offerts par ces réseaux et les modes de communication existants.

1. Définition des réseaux

a. Réseau Ad-hoc

Les réseaux Ad-hoc sont des réseaux sans-fil capables de s'organiser spontanément et de manière autonome dans leur environnement [5]. Les nœuds proches effectuent une communication directe entre eux, tandis que les autres nœuds jouent le rôle d'intermédiaire pour transporter le message jusqu'à sa destination finale. Dans le cas du réseau VANET, les véhicules sur la route jouent le rôle des nœuds qui transmettent le message dans les protocoles de routage.

Dans un réseau Ad-hoc on rencontre plusieurs problèmes comme l'absence d'infrastructure, une bande passante limitée, beaucoup de perte de données, des erreurs de transmission, et plusieurs problèmes de sécurité.

b. Réseau VANET

Un réseau VANET est un réseau de communication entre véhicules intelligents équipés de calculateurs, de périphériques réseau et de différents types de capteurs.

Les VANETs font partie de la famille des réseaux mobiles MANET qui fonctionnent dans des réseaux à liaison pair à pair sans infrastructure, c'est-à-dire que tout nœud appartenant au réseau pourra être considéré comme un point d'accès. Dans un réseau véhiculaire, les nœuds sont des véhicules intelligents appartenant au réseau et qui peuvent communiquer entre eux directement s'ils sont à portée ou via des véhicules intermédiaires s'ils sont éloignés. [6]

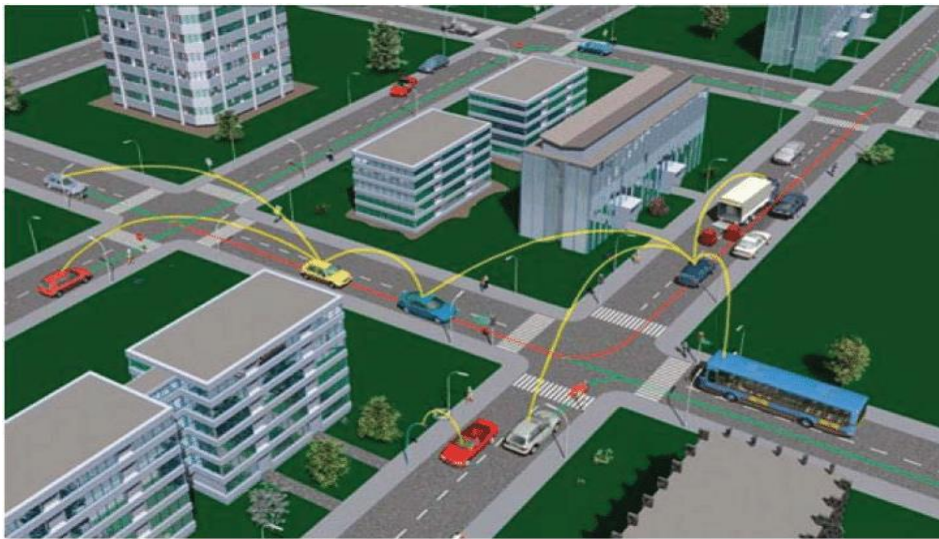


Figure 1.1 : Exemple d'un réseau VANET.

2. Architecture du réseau VANET

a. Entités de communication

i. Road Side Unit

Ces entités sont les subordonnés des CA. Elles sont installées au bord des routes. Elles peuvent être principalement, des feux de signalisation, des lampadaires ou autres. Leur principale responsabilité est de soutenir la TA dans la gestion du trafic et des véhicules. Elles représentent des points d'accès au réseau et aux différentes informations sur la circulation. [7]

ii. Autorité centrale

C'est une source d'authenticité de l'information. Elle assure la gestion et l'enregistrement de toutes les entités sur le réseau (RSU et OBU). La TA est sensée connaître toutes les vraies identités des véhicules et au besoin les divulguer pour les forces de l'ordre.

Aussi, la TA dans certains travaux se charge de la délivrance et l'attribution des certificats et des pseudonymes de communications. [7]

iii. On Board Unit

Ce sont des unités embarquées dans les véhicules intelligents, elles regroupent un ensemble de composants matériels et logiciels de hautes technologies (GPS, radar, caméras, différents capteurs et autres). Leurs rôles sont d'assurer la localisation, la réception, le calcul, le stockage et l'envoi des données sur le réseau. Ce sont des émetteurs-récepteurs qui assurent la connexion du véhicule au réseau. [7]

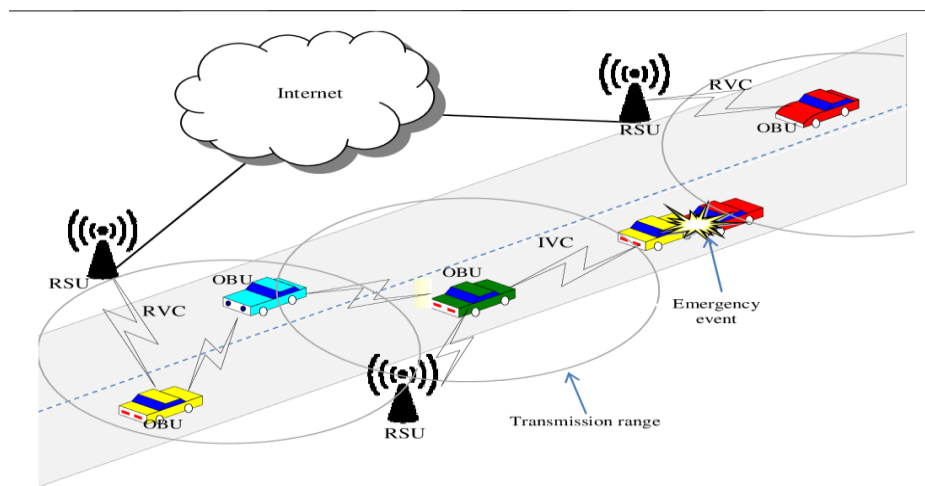


Figure 2.2: les entités de communication.

b. Modes de communication

Dans les réseaux VANETs, on trouve principalement, les entités fixes qui constituent l'infrastructure (RSU et TA) et les entités mobiles (les véhicules). Pour pouvoir échanger les différentes informations et données liées à la sécurité et au confort des usagers de la route, ces différentes entités doivent établir des communications entre elles. Pour cette raison, on distingue trois types de communications véhicule à véhicule (V2V) et véhicule à infrastructure (V2I) et hybride. [8]

i. Communication en mode Ad-hoc (V2V)

Dans ce mode, un réseau de véhicules est vu comme un cas particulier de réseaux mobiles ad-hoc MANET où les contraintes d'énergie, de mémoire et de capacité de calcul sont

relaxées et où le modèle de mobilité n'est pas aléatoire, mais prévisible avec une très forte mobilité. Cette architecture peut être utilisée dans les scénarios de diffusion d'alerte (freinage d'urgence, collision, ralentissement, etc.) ou pour la conduite coopérative. En effet, dans le cadre des applications de la sécurité routière, les réseaux à infrastructure montrent leurs limites, surtout en termes de délais. Et il est clair qu'une communication ad-hoc multi-sauts est plus performante qu'une communication passant par un réseau d'opérateur ou une infrastructure. La figure II.3 illustre ce mode de communication entre véhicules. [9]

ii. Communication en mode infrastructure (V2I)

Ce mode de communication permet une meilleure utilisation des ressources partagées et démultiplie les services fournis (exemple d'échange de données de voiture à domicile, accès Internet, diagnostic distant pour réparer un véhicule, etc.) grâce à des points d'accès RSU déployés aux bords des routes. Ce mode est inadéquat pour les applications liées à la sécurité routière car les réseaux à infrastructure ne sont pas performants quant aux délais d'acheminement. [10]

iii. Communication en mode hybride

Ce type de communication est une combinaison des deux modes de communications précédentes V2V et V2I, et comme les portées des infrastructures étant limitées alors que l'utilisation des véhicules comme relais permet d'étendre cette distance et d'éviter la multiplication des stations de base à chaque coin de la route et cela conduit à une communication hybride très intéressante et économique. [9]

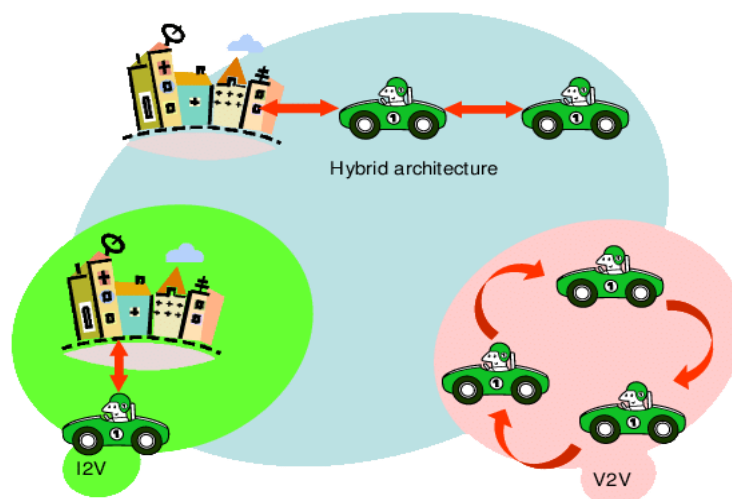


Figure 3.3 : les modes de communication.

c. Types de messages

i. Messages de contrôle

Ce type de message contient souvent des informations relatives à l'identité et à l'état actuel du véhicule (Position, vitesse, direction et autres), il est diffusé périodiquement et est utilisé principalement pour permettre l'identification du voisinage. Ce type de message joue un rôle primordial dans la plupart des protocoles de routage et de sécurité. [11]

ii. Messages d'alerte

Ce type de message est envoyé pour prévenir les autres véhicules de différentes urgences et des catastrophes sur la route (accident, congestion de la circulation, information météorologique, passage d'un véhicule de secours et autres), afin qu'ils aient plus de temps pour agir. Ce type de message aide à améliorer la circulation et la sécurité routière. [11]

3. Les caractéristiques des VANETS

Bien que certaines des caractéristiques des VANETs ressemblent aux caractéristiques des MANETs, mais il existe des caractéristiques spécifiques pour les VANETs qui peuvent être classées comme suit : [12]

a) Topologie hautement dynamique

La vitesse élevée des véhicules ainsi que la disponibilité de choix de chemins multiples caractérisent la topologie dynamique des VANETs.

b) Mobilité de la modélisation et de la prévention

La prédiction de la position du véhicule et de ses mouvements est très difficile. Par ailleurs, la modélisation de la mobilité et la prédiction dans les VANETs sont basées sur la disponibilité des modèles de feuille de route déjà prédéfinis. La vitesse des véhicules est considérée comme une métrique importante pour une conception d'un réseau efficace.

c) Environnement de communication

Le modèle de mobilité peut avoir différentes caractéristiques selon l'architecture routière, les autoroutes ou les environnements urbains. A cet effet, la communication dans ces situations doit être prise avec soin.

d) Contrainte de délai difficile

Au moment de l'urgence, la remise des messages en fonction du temps devient un problème critique. Par conséquent, gérer de telles situations devient primordial que se contenter au taux de données élevés.

e) Interaction avec les capteurs embarqués

Les capteurs peuvent lire les données relatives à la vitesse du véhicule, direction et peuvent communiquer avec le centre de données. Ainsi, les capteurs peuvent être aussi utilisés dans la formation de liens et dans les protocoles de routage.

f) Batterie et stockage illimités

Les nœuds dans les VANET ne souffrent pas d'alimentation et de limitation de stockage comme dans les réseaux de capteurs, ainsi le cycle de service n'est pas aussi pertinent que dans les réseaux de capteurs.

4. Les services offerts par les réseaux VANET

a. Les services liés à la sécurité routière

Ces services concernent les applications ayant un impact direct sur la sécurité des personnes et des biens, c'est-à-dire les applications qui permettent de réduire le nombre des accidents routiers et d'améliorer les conditions de circulation.

Les services liés à la sécurité routière se basent sur la détection de l'environnement proche au moyen de capteurs (par exemple : les radars et les caméras) installés au niveau des véhicules ou bien au centre de contrôle, ainsi que la diffusion de messages fournissant des informations sur l'état du réseau routier (trafic, travaux, météo), ou rappelant au conducteur les limitations de vitesse, les distances de sécurité ou qu'il s'approche d'une intersection, avant même de la voir. [13]

b. Les services liés au confort

Les réseaux VANET ne se contenteront pas seulement à offrir des services liés à la sécurité des véhicules et leurs occupants, mais permettront aussi d'assurer le confort durant leurs voyages ; ces services comprennent par exemple : la messagerie instantanée, les jeux en réseau, l'accès à Internet, les paiements automatiques et la diffusion d'informations utiles sur la disponibilité de l'espace de stationnement dans les parkings en indiquant aux conducteurs les espaces libres. Le champ d'application de ces services, à ce stade, est très large et offre des perspectives intéressantes aux opérateurs de télécommunications en leur permettant de réaliser des bénéfices supplémentaires.

5. Les applications des VANETS

Selon le DSRC "Dedicated Short Range Communications", il ya plus d'une centaine de demandes recommandées dans VANETs. Ces applications sont de deux catégories, la sécurité et non liées à la sécurité. En outre, ils peuvent être classés en OBU à OBU ou applications OBU à RSU. Voici une liste de certaines de ces applications. [14]

* Avertissement coopératif de collision :

Avertissement coopératif de collision est une application de sécurité OBU à OBU, c'est – à-dire, en cas de tout changement brusque de vitesse ou de direction de conduite, le véhicule est considéré comme anormale et diffuse un message d'avertissement pour avertir tous les véhicules suivants du danger probable. Cette application nécessite un algorithme de diffusion efficace avec une très faible latence.

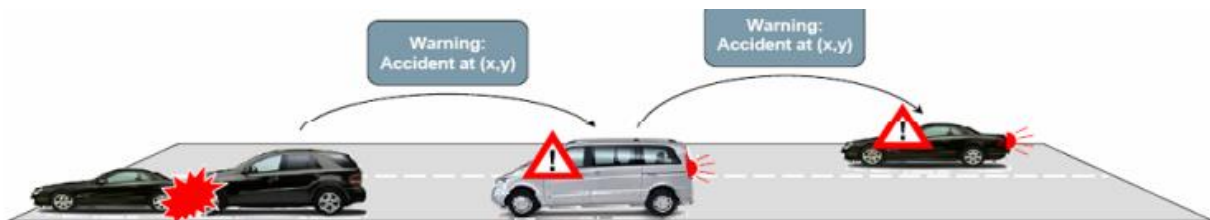


Figure 4.4 : réduire les accidents.

* Avertissement de changement de ruelle :

Avertissement de changement de ruelle est une application de sécurité OBU à OBU, c'est-à-dire un conducteur de véhicule peut avertir d'autres véhicules de son intention de modifier la voie de déplacement et pour réserver un couloir vide dans la voie approchée. Cette application dépend de radiodiffusion.

* Avertissement de collision d'intersection :

Avertissement de collision d'intersection est une application de sécurité OBU à RSU. Aux intersections, un nœud centralisé avertit les véhicules qui s'approchent des accidents possibles et les aides à déterminer la vitesse d'approche appropriée. Cette application n'utilise que des messages de diffusion.

* L'approche de véhicules de secours :

L'approche de véhicules de secours est une application de sécurité publique OBU à OBU. Les véhicules d'urgence à grande vitesse (ambulance ou une voiture de Police) peuvent Avertir les autres véhicules pour dégager leur ruelle. Encore une fois, cette application dépend sur la radiodiffusion.

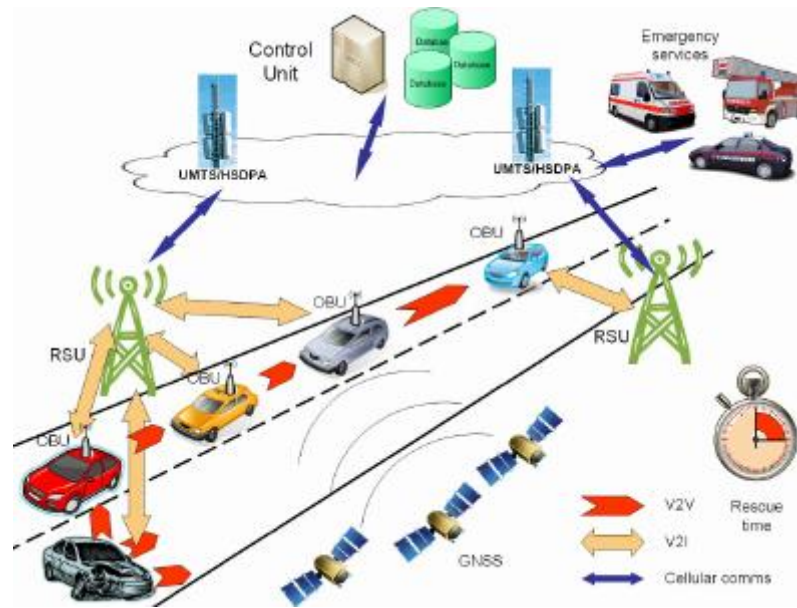


Figure 5.5 : Notifications des accidents.

* **Avertissement de renversement :**

Avertissement de renversement est une application de sécurité OBU à RSU. Un RSU localisé à courbes critiques peut diffuser des informations sur l'angle de la courbe et l'état de la route, de sorte qu'en s'approchant de véhicules, peut déterminer la vitesse maximale possible avant le retournement.

* **Avertissement de domaine d'activités :**

Avertissement de domaine d'activités est une application de sécurité OBU à RSU. Un RSU est monté dans les zones de travail pour avertir les véhicules entrants du danger probable et les mettre en garde pour diminuer la vitesse et de changer la voie de circulation.

* **Communications inter-véhicule :**

La communication inter-véhicules est une application OBU à OBU non sécurisée qui permet aux voyageurs de communiquer entre eux en utilisant le transfert de fichier instantané, la vidéo conférence ou même le chat vidéo.

* **Collection électronique de péage :**

Télépéage est une application OBU à RSU non sécurisée qui prend en charge la collecte de paiement aux gares de péage, en utilisant des systèmes automatisés pour accroître l'efficacité opérationnelle. Les systèmes sont généralement constitués d'OBU qui est à la charge des cartes à puce prépayées. Ces OBU sont identifiés par UAR situés dans des voies réservées aux postes de péage.

* **Paiement de parking :**

Le paiement de parking est une application OBU à RSU non sécurisée qui fournit des avantages pour les exploitants de parcs de stationnement, simplifier le paiement pour les clients et de réduire la congestion aux entrées et sorties de parcs de stationnement.

* **Gestion du trafic :**

Dans le véhicule de navigation est une application non sécurisée qui est conçu pour réduire le temps de conduite et la consommation de carburant en échangeant des informations en temps réel sur les conditions de circulation dans la voie de conduite.

* **Couplage / découplage :**

Système de couplage / découplage est une application OBU à OBU non sécurisée qui vise à relier plusieurs bus ou camions dans un train, afin de minimiser le temps de distance de progrès et de voyage et de diminuer les collisions arrières.

6. Classification des protocoles de routage dans les réseaux VANET

Les réseaux véhiculaires ont comme caractéristique principale une forte mobilité qui entraîne une topologie très dynamique. Cette caractéristique fait que les protocoles de routage traditionnels des MANETS sont pour la plupart inadaptée aux VANETs. En effet, dans les VANETs, la vitesse peut être beaucoup plus élevée que les MANETS dans certains environnements de communication comme les autoroutes. Différentes solutions ont été proposées pour le routage dans les réseaux VANETs, nous distinguons deux classes : les protocoles basés sur la Unicast (topologie) qui sont divisés en protocoles proactifs, réactifs et hybrides pour créer des itinéraires et les protocoles basés sur la localisation (géographique) qui utilisent la position physique des nœuds mobiles pour configurer le routage. [15]

a. Le routage basé sur la topologie

i. Les protocoles proactifs

Le principe de base des protocoles proactifs est de calculer les routes à l'avance, en continu ainsi de maintenir à jour les tables de routage, de tel sorte que lorsqu'un nœud désire envoyer un paquet à un autre nœud, une route soit immédiatement connue. Les nœuds dans les réseaux mobiles Ad Hoc peuvent apparaître et disparaître de manière aléatoire d'où la nécessité de mise en place d'un système d'échange continuels des paquets de contrôle. Les

tables de routage sont donc maintenues grâce à ces paquets. Cette manière de procéder permet aux noeuds de construire de façon distribuée la topologie du réseau, lorsqu'un noeud reçoit un paquet de contrôle, il met à jour ses tables de routage. Ainsi, de nouvelles routes seront construites sur la base des informations topologiques transportées par les trames de contrôle. Ce processus est déclenché à chaque changement de topologie pour reconstruire à nouveau les routes. Un des avantages de ces protocoles est la disponibilité immédiate de la route lors du besoin. Cependant, la bande passante diminue par le trafic généré par l'échange de paquets de contrôle. [16,17]

*** Le protocole FSR (Fisheye State Routing)**

Il est similaire au Protocole de liaison de l'état de routage (LSR). Chaque nœud maintient une table de topologie basée sur les dernières informations reçues des nœuds de quartier. Il utilise période d'échange différent pour différentes entrées dans la table de routage afin de réduire la taille des messages de commande dans les grands réseaux. L'inconvénient du FSR, est que la taille de la table de routage augmente avec l'augmentation de la taille du réseau, la découverte de route peut échouer si le nœud de destination se trouve hors de portée du nœud de source. En raison de la grande mobilité dans VANET, la route vers la destination à distance devient moins précise. [18]

*** Le protocole OLSR (Optimized Link State Routing Protocol)**

C'est un protocole de routage destiné aux réseaux maillés, sans fil ou mobiles. Le protocole est une optimisation de l'algorithme d'état de liaison pure. Le concept clé utilisé dans le protocole est l'utilisation des relais multipoints (MPR). L'ensemble MPR est choisi de sorte qu'il couvre tous les nœuds qui sont à deux sauts de suite, Il fonctionne comme un protocole proactif, les informations de topologie avec d'autres nœuds du réseau sont échangées régulièrement. [19]

ii. Les protocoles réactifs

Le principe des protocoles réactifs également appelés protocoles de routage à la demande est de lancer le processus de recherche de routes uniquement en cas de besoin, cela permet d'économiser de la bande passante et de l'énergie. Lorsqu'un paquet doit être envoyé d'un nœud source vers un nœud cible, le protocole de routage va rechercher un chemin jusqu'à la destination, une fois ce chemin trouvé, il est inscrit dans la table de routage et peut être utilisé tant que la destination est joignable ou jusqu'au moment où la route devient inutile. En général, cette recherche se fait par inondation (un paquet de recherche de route est

transmis de proche en proche dans tout ou partie du réseau). Dans ce cas, la bande passante est plus large. Cependant, du fait que l'on ne dispose pas immédiatement de la route vers la destination, le délai d'établissement de la route est plus important en comparaison avec les protocoles proactifs. [16,17]

* **Le protocole AODV (Ad hoc On Demand Distance Vector)**

C'est un protocole de routage destiné aux réseaux ad hoc. Il est à la fois capable de routage unicast et multicast. Il est libre de boucle, auto démarrant et s'accommode d'un grand nombre de nœuds mobiles.

Lorsqu'un nœud source demande une route, il crée les routes à la volée et les maintient tant que la source en a besoin. Pour les groupes multicast, AODV construit une arborescence.

Ce protocole de routage est peu gourmand en énergie et ne nécessite pas de grande puissance de calcul, il est donc facile à installer sur de petits équipements mobiles. [20]

* **Le protocole DSR (Dynamic Source Routing)**

C'est un protocole de routage pour les réseaux maillés sans fils (wireless mesh networks). Il est similaire à AODV dans le sens qu'il forme une route à la demande, quand un ordinateur veut transmettre. Cependant, il utilise le routage à la source au lieu de se baser sur la table de routage de chaque routeur intermédiaire. [20]

iii. Les protocoles hybrides

Les protocoles hybrides combinent les deux approches précédentes. Ils utilisent une technique proactive dans un petit périmètre autour de la source où le nombre de sauts est assez petit (par exemple trois à quatre sauts) et réactive pour les nœuds plus éloignés. L'avantage des protocoles hybrides est le fait qu'ils s'adaptent mieux aux réseaux de grandes tailles. Cependant, ce type de protocole cumule les inconvénients des protocoles proactifs et ceux des protocoles réactifs, tels que l'échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un nœud éloigné. [16,17]

* **Le protocole ZRP (Zone Routing Protocol)**

C'est un modèle hybride entre un schéma proactif et un schéma réactif. Le principal problème dans l'élaboration d'un protocole de routage pour réseau ad hoc réside dans le fait que pour déterminer le parcours d'un paquet de données, le nœud source doit au moins connaître les informations permettant d'atteindre ses proches voisins. D'un autre côté, la topologie d'un tel réseau change fréquemment. De plus, comme le nombre de nœuds peut être élevé, le nombre

de destinations potentielles peut également l'être, ce qui requiert des échanges de données important et fréquents. Donc la quantité de données de mise à jour du trafic peut être conséquente. Cela est en contradiction avec le fait que toutes les mises à jour dans un réseau interconnecté ad hoc circulent dans l'air et donc sont coûteuses en ressources. [21]

* **Le protocole HARP**

Il divise l'ensemble du réseau en zones non-chevauchées. Il vise à établir un itinéraire stable depuis une source vers une destination à améliorer le retard. Il s'applique dans la découverte de route entre les zones pour limiter les inondations dans le réseau et de choisir le meilleur itinéraire en fonction des critères de stabilité.

HARP est effectué à deux niveaux : intra-zone et inter-zone en fonction de la position de destination, Il utilise des protocoles proactifs et réactifs en intra-zone et inter-zone routage respectivement. Il n'est pas applicable dans les réseaux ad hoc haute mobilité. [22]

b. Les protocoles de routage basés sur la géographie

Les protocoles de routage géographiques sont les plus adaptés pour les réseaux ad hoc de véhicules, puisque le mécanisme de routage se base sur les données géographiques des nœuds. [23]

* **Le protocole GSR (Geographic Source Routing)**

Le GSR est un protocole de routage géographique qui combine entre le routage basé sur la position et le routage basé sur la topologie. Son routage est basé sur la topologie car il calcule et stocke l'image de la topologie générale du réseau actuel et le met à jour périodiquement en cas de déplacement des nœuds du réseau, à partir de cette image et des informations géographique (routage basé sur la position) il assure l'acheminement le plus court à l'aide aussi des algorithmes de recherches.

* **Le protocole A-STAR (Anchor-based Street and Traffic Aware Routing)**

C'est un Protocol basé sur la position pour un environnement de communication intervéhiculaire métropolitain, Il utilise particulièrement les informations sur les itinéraires d'autobus de ville pour identifier une route d'ancre (Anchor route) avec une connectivité élevée pour l'acheminement des paquets. A-STAR est similaire au protocole GSR en adoptant une approche de routage basée sur l'ancre (Anchorbased) qui tient compte des caractéristiques des rues. Cependant, contrairement à GSR il calcule les "Anchor Paths" en fonction du trafic (trafic de bus, véhicules, etc.). Un point est assigné à chaque rue en fonction de sa capacité.

Nous trouvons qu'une des perspectives à ce travail consiste à donner un poids dynamique et qui changerait en fonction de ces informations collectées et du trafic à un moment donné, afin de fournir une meilleure qualité de calcul d'ancres.

* **Le protocole Gy-TAR (Greedy Traffic-Aware Routing Protocol)**

Le protocole de routage Gy-TAR est un protocole géographique basé sur la position et adapté aux réseaux véhiculaires, capable de trouver des chemins robustes dans un environnement urbain. Chaque véhicule connaît sa propre position grâce à l'unité du système de positionnement global (GPS) et l'émetteur connaît la position de destinataire grâce à des services de localisation tel que GLS (Grid Localisation Service) et peut déterminer la position des nœuds voisins à un saut à travers des cartes numériques.

* **Le protocole DVCAST (Distributed Vehical Broadsact Protocol)**

C'est un protocole basé sur la diffusion distribué, il divise les véhicules en trois catégories selon leur connectivité locale : connectés, peu connectés/zone de voisinage ou bien totalement déconnectés. Chaque véhicule utilise un drapeau (flag) pour vérifier si le paquet est redondant ou non en utilisant les informations locales de la topologie en inondant le réseau par des messages périodiques « HELLO ».

* **Protocole à base de clusterisation**

Dans les protocoles de routage véhicules à base de cluster, près les uns des autres forment un cluster. Chaque groupe a un cluster-head (grappe tête), qui est responsable des fonctions de gestion pour intra et inter-clusters. Les nœuds intra-cluster communiquent entre eux en utilisant les liens directs, alors inter-cluster communication est réalisée via cluster head. [24]

* **Le protocole HCB (Hard Core Bitch)**

C'est un protocole de routage basé sur cluster hiérarchique conçu pour les réseaux ad hoc hautement mobile. Son architecture de communication est basée sur deux couches. Dans la couche 1 pour la plupart des nœuds ont une interface radio unique et ils communiquent les uns avec les autres par la voie multi-hop. Parmi ces nœuds certains ont également une autre interface avec la gamme de communication à longue radio appelée super nœuds, qui existent à la fois sur la couche 1 et 2. Super nœuds sont capables de communiquer les uns avec les autres via la station

de base en couche 2. Au cours de la formation de grappes, chaque nœud sera privilégié la plus proche tête de cluster et des super nœuds deviendront pôle têtes de couche 1.

Dans HCB, le routage intra-cluster est effectué indépendamment dans chaque grappe.

7. Conclusion

Dans ce chapitre, nous avons présenté les réseaux véhiculaires VANETs, ses modes de communication, caractéristique, applications, et la classification des protocoles de routage dans les réseaux VANET.

Dans la suite de notre travail, nous allons s'intéresser aux réseaux de Petri en général, aussi les nets within nets.

Chapitre III Analyse et conception

SECTION 1 : OUTILS DE CONCEPTION

1. Introduction

Les réseaux de Petri constituent un modèle fondamental pour la modélisation de systèmes discrets (systèmes informatiques, systèmes de fabrication, systèmes de communication, etc...). Ils ont été inventés par Carl Adam Petri et utilisés dans des nombreux domaines, et sont adaptés à la modélisation de systèmes de production, de transport et de communication.

Dans cette section Nous présentons dans un premier lieu les RdP généralisés. Ensuite, nous parleront des systèmes multi-agents mobiles et en particulier les réseaux de référence. À la fin en présentent le MULAN (Multi-Agent Nets architecture).

2. Les Réseaux de Petri

Ces réseaux présentent des caractéristiques intéressantes telles que la modélisation et la visualisation de comportements parallèles, de la synchronisation et partage de ressources. De plus leurs aspects théoriques ont été largement étudiés et les résultats théoriques les concernant sont très abondant. [25]

Et les principaux utilisateurs de ces réseaux sont les informaticiens et les automaticiens. Cependant c'est un outil assez général pour modéliser des phénomènes très variées. Il permet notamment :

- La modélisation des systèmes informatiques.
- L'évaluation des performances des systèmes discrets, des interfaces homme-machine.
- La commande des ateliers de fabrication.
- La conception de systèmes temps réel.
- La modélisation des protocoles de communication.
- La modélisation des chaînes de production (de fabrication).

Les atouts des RdP :

- Ils permettent de d'écrire de manière précise mais non formelle la structure d'un système.
- Ils offrent un support graphique de conception.

- Ils permettent de décrire un système étape par étape, en décomposant en éléments plus simples les éléments constitutifs initiaux du système.
- Ils permettent de décrire à l'aide d'un même support de base, à la fois la structure et la dynamique d'un système.
- Ils permettent de passer d'une description graphique d'un système à une description formelle permettant l'analyse mathématique du système (cohérence).

a. Concepts de base

* **Condition :**

Une condition est un prédicat logique d'un état du système. Elle est soit vraie, soit fausse.

Modélisée à l'aide d'une place.



* **Événement :**

Les événements sont des actions se déroulant dans le système. Le déclenchement d'un événement dépend de l'état du système.

Un état du système peut être décrit comme un ensemble de conditions.

Modélisé à l'aide d'une transition.

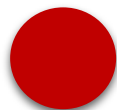


* **Déclenchement, pré-condition, post-condition:**

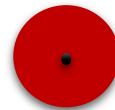
Les conditions nécessaires au déclenchement d'un événement sont les pré-conditions de l'événement.

Lorsqu'un événement se produit, certaines de ses pré-conditions peuvent cesser d'être vraies alors que d'autres conditions, appelées post-conditions de l'événement deviennent vraies.

Satisfaction d'une condition : modélisée à l'aide d'un jeton.



Condition fausse



Condition vraie

b. Définition formelle

Formellement un RdP est un quintuple, $R = (P, T, F, W, M_0)$ tel que :

- $P = \{p_1, p_2, \dots, p_m\}$ ensemble fini de m places ;
- $T = \{T_1, T_2, \dots, T_n\}$ ensemble fini de n transitions ;
- $F \in (P \times T) \cup (T \times P)$ ensemble d'arcs;
- $W : F \rightarrow \{1, 2, 3, \dots\}$ fonction de poids ;
- $M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$ marquage initial ;

Le RdP dont tous ses arcs sont de poids "1" est appelé RdP ordinaire.

Dans le cas où les arcs peuvent avoir des poids supérieurs à "1", il s'agit de RdP généralisé. La structure de RdP est donnée par le quadruplet [26] $Q = (P, T, F, W)$, Q représente le RdP sans aucune spécification de marquage initial M_0 . Un RdP marqué est noté par $R = (Q, M_0)$.

Soit $R = (P, T, F, W, M_0)$ un RdP. On a les notations suivantes:

- ${}^\circ t$: l'ensemble des places d'entrée de la transition t .
- t° : l'ensemble des places de sortie de la transition t .
- ${}^\circ p$: l'ensemble des transitions d'entrée de la place p .
- P° : l'ensemble de transitions de sortie de la place p .

c. Définition informelle

Informellement un RdP est un graphe orienté composé de 2 types de nœuds :

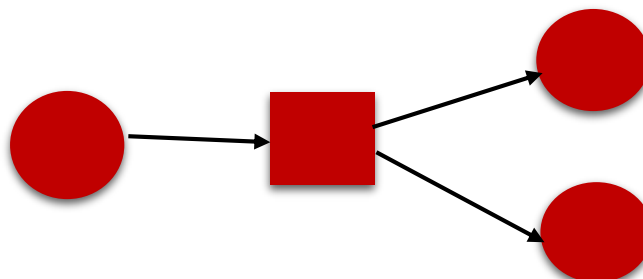
- Les places (P_i) qui permettent de décrire les états du système modélisé. L'ensemble de ces places est noté $P = \{P_1, P_2, \dots\}$.

- Les transitions (T_i) qui représentent les changements d'états.

L'ensemble de ces transitions est noté $T = \{T_1, T_2, \dots\}$.

- Les Places et transitions sont reliées par des arcs orientés. On dira qu'un RdP est un graphe biparti orienté.

A chaque arc, on attribue un poids (nombre entier). Par défaut ce nombre est égal à 1.

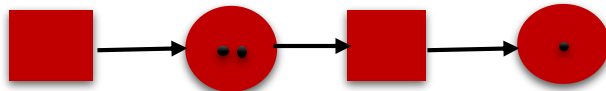


Exemple 1

d. Marquage

Chaque place contient un nombre entier (positif ou nul) de marques ou jetons. Le nombre de marque contenu dans une place P_i sera noté soit $M(P_i)$ soit m_i . Le marquage du réseau à l'instant i , M_i est défini par le vecteur de ces marquages m_i c'ad $M_i = (m_1, m_2, \dots, m_n)$.

Le marquage dit initial décrit l'état initial du système modélisé (M_0). [27]



Exemple 2

Ce RdP possède 2 places, 2 transitions et 3 arcs orientés. Soit donc :

Le marquage initial est $M_0 = (2,1)$

e. Franchissement d'une Transition :

Pour rendre compte de l'évolution du système modélisé, les réseaux de Petri intègrent un formalisme permettant de passer d'un marquage à un autre : c'est le franchissement des transitions.

Le franchissement (ou le tir) d'une transition ne peut s'effectuer que si chacune des places en amont (en entrée) de cette transition contient suffisamment de jetons (\geq au poids de l'arc correspondant). On dit alors que la transition est franchissable ou validée.

Le franchissement est une opération indivisible (atomique) qui consiste à retirer des jetons des places en amont (en entrée) et à ajouter des jetons dans les places en aval (en sortie) de la transmission franchie.

Le nombre de jetons retirés ou ajoutés est égal au poids de l'arc reliant la transition à la place en question. [28]

f. Propriétés des Rdp

1. Bornitude

Un RdP est borné si et seulement si toutes ses places sont bornées. Une place p est bornée si pour tout marquage accessible M , le nombre de jetons dans la place est inférieur à une constante k .

Le caractère borné d'un RdP renseigne sur les valeurs limites des ressources demandées par le système. Si un réseau est borné et la borne est égale à un, alors il est dit sauf.

2. Vivacité

L'évolution du marquage d'un RdP se fait par franchissement de transitions. Lorsqu'au cours de son évolution, certaines transitions ne sont jamais franchies, cela indique que l'évènement associé à la transition ne se produit pas et que le marquage d'une partie du RdP n'évolue pas. Cela indique que le sous-système modélisé par cette partie-là ne fonctionnera pas. Il y a donc un problème au niveau de la conception du système. L'idée est d'être capable de détecter systématiquement ce phénomène par l'analyse de propriétés du modèle RdP du système afin de disposer d'un outil d'aide à la conception des systèmes.

Une transition t est vivante pour un marquage initial M_0 si pour tout marquage M accessible à partir de ce marquage initial, il existe une séquence de franchissements à partir de M contenant t . Autrement dit, quelle que soit l'évolution, il existera toujours une possibilité d'arriver à franchir t .

Un RdP est vivant pour un marquage initial M_0 si toutes ses transitions sont vivantes pour ce marquage initial. Dans un tel réseau, on garantit que chaque transition est franchissable éventuellement peu importe l'état du système. La propriété de vivacité est une propriété forte, souvent difficile à vérifier. [26] [29]

3. Blocage :

Un blocage est un marquage tel qu'aucune transition n'est validée. Un RdP est dit donc sans blocage (pseudo vivant) pour un marquage initial M_0 si aucun marquage accessible M_i n'est un blocage.

4. Réinitialiabilité :

Un RdP est réinitialisable (ou réversible) si et seulement si pour tout marquage M , il existe une séquence de transitions qui permet de revenir au marquage initial M_0 . Cette propriété renseigne sur le fonctionnement répétitif, ce qui est pertinent pour la majorité des systèmes interactifs pratiques.

g. Les méthodes d'analyse des réseaux de Petri

On aborde deux méthodes d'analyse : graphe de marquages et équation de matrice. La première méthode est de construire le graphe de tous les marquages de réseau, et on déduit des propriétés

grâce aux techniques de théorie de graphe. La deuxième méthode est de trouver une représentation de matrice de réseau, et on utilise des techniques d'algèbre linéaire pour obtenir des propriétés du réseau.

La vérification des propriétés dynamiques se fait généralement en établissant l'arbre de couverture (ou le graphe de couverture). L'arbre de couverture est composé de nœuds qui correspondent aux marquages accessibles, et d'arcs correspondant aux franchissements de la transition qui fait passer d'un marquage à l'autre. Le nombre de nœuds dans cet arbre est fini. La construction de l'arbre de couverture permet de décider si un RdP est borné : on dit que la propriété de réseau borné est décidable. Dans ce cas, l'arbre sera appelé le graphe des marquages accessibles. A partir de ce graphe, on peut vérifier toutes les autres propriétés (vivacité, accessibilité d'un marquage, ...). Cependant, pour un RdP non borné, établir l'arbre de couverture ne suffit pas pour résoudre le problème d'accessibilité et le problème de vivacité. Ces deux problèmes sont également décidables, mais par des algorithmes plus compliqués. Malgré ceci, le graphe de couverture reste le seul outil pour vérifier les propriétés dynamiques d'un RdP pour un marquage initial donné.

L'analyse des propriétés structurelle repose essentiellement sur les techniques d'algèbre linéaire. En fait, la structure d'un RdP peut être définie par une matrice à éléments entiers, appelée matrice d'incidence. [30]

h. Quelques extensions des réseaux de Petri

Les RdP permettent de modéliser un certain nombre de comportements importants dans les systèmes : le parallélisme, la synchronisation, le partage de ressources... etc.

Cependant cette modélisation souffre d'une limitation. En effet, la modélisation des systèmes réels mène parfois à des RdP de taille ingérable, ce qui diminue leur lisibilité et rend leur manipulation et même leur analyse difficile (problème d'explosion combinatoire). Pour apporter des solutions à cette limitation et augmenter la puissance de modélisation, des extensions du modèle RdP dites des RdP de haut niveau ont été proposées dans lesquelles d'autres aspects ont été pris en compte, tel que ; la colorisation, la temporisation, et autres. Dans ce qui suit nous explorons quelques extensions des RdP.

1. Réseaux de Petri colorés

Les réseaux de Petri colorés sont des réseaux de Petri dans lesquels les jetons portent des couleurs. Une couleur est une information attachée à un jeton. Cette information permet de distinguer des jetons entre eux et peut-être de type quelconque. Ainsi, les arcs ne sont pas

seulement étiquetés par le nombre de jetons mais par leurs couleurs. Le franchissement d'une transition est alors conditionné par la présence dans les places en entrée du nombre de jetons nécessaires, qui en plus satisfont les couleurs qui étiquettent les arcs. Après le franchissement d'une transition, les jetons qui étiquettent les arcs d'entrée sont retirés des places en entrée tandis que ceux qui étiquettent les arcs de sortie sont ajoutés aux places en sortie de cette transition. Les réseaux colorés n'apportent pas de puissance de description supplémentaire par rapport aux réseaux de Petri, ils permettent juste une condensation de l'information. [31]

2. Réseaux de Petri temporisés

Aucune durée n'est liée au franchissement des transitions et/ou au temps de séjour des marques dans les places en ce qui concerne les RdP. Cependant il y a beaucoup de systèmes à événements discrets dont l'évolution dépend du temps. D'autre part la notion de temps est capitale lorsque l'on veut évaluer les performances ou étudier les problèmes d'ordonnancement d'un système dynamique. La nécessité de modéliser et d'étudier de tels systèmes a donné naissance aux RdP temporisés (RdPT).

Le premier travail sur les RdPT a été réalisé par Ramchandani. Son modèle (les réseaux de Petri t-temporisés (t-RdPT)) associe à chaque transition d'un RdP autonome un paramètre temporel dont la sémantique correspond à la durée de tir de cette transition. Un autre modèle a été proposé plus tard par Sifakis, : les réseaux de Petri p-temporisés (p-RdPT). Un paramètre temporel est associé à chaque place. La sémantique de ce paramètre correspond au temps de séjour minimum d'une marque dans une place (temps d'indisponibilité). Sifakis montrera plus tard, que son modèle et celui de Ramchandani sont équivalents. [30]

3. Réseaux de Petri synchronisés

Dans les modélisations RdP que nous avons vues précédemment, le fait qu'une transition soit franchissable indique que toutes les conditions sont réunies pour qu'elle soit effectivement franchie. Le moment où se produira le franchissement n'est pas connu. Un RdP synchronisé est un RdP où à chaque transition est associée un événement. La transition sera alors franchie si elle est validée mais quand l'événement associé se produit.

Dans un RdP synchronisé, une transition validée n'est pas forcément franchissable. La transition est validée quand la condition sur les marquages est satisfaite. Elle deviendra franchissable quand l'événement externe associé à la transition se produit : elle est alors immédiatement franchie. Si en fonction du marquage de ses places d'entrée, plusieurs

franchissements sont possibles, un seul se produira effectivement, celui dont l'événement associé se produit en premier. [31]

3. Le paradigme (nets-within-nets)

Le paradigme des réseaux dans les réseaux (nets-within-nets) selon Valk, constitue la suite logique de ses travaux antérieurs portant sur les réseaux de flux de tâches (task-flow nets). Ce paradigme formalise le fait que les jetons dans un réseau de Pétri peuvent symboliser une structure de donnée complexe voire même un réseau de Petri.

Partant de ce principe, il est possible de modéliser des structures hiérarchiques d'une manière relativement simple mais très élégante. Dans ce qui suit, nous allons donner une brève introduction d'une implémentation de certains aspects des réseaux dans les réseaux appelés les réseaux de référence. [32]

a. Les réseaux de référence

Les réseaux de référence [33] sont une notation graphique très appropriée à la description et à l'exécution des processus concurrents et complexes. Telle la majorité des autres formalismes Petri, il existe un outil de simulation de ces réseaux appelé RENEW (Reference net workshop). Les réseaux de référence étendent les réseaux de Petri classiques et les réseaux de Petri colorés par l'introduction de nouvelles notions telles que : les instances réseaux, les réseaux considérés comme jetons objets, la communication via des canaux synchrones et l'utilisation de différents types d'arcs. En dépit de tout ceci, ces réseaux restent très similaires aux réseaux de Petri colorés tels qu'ils ont été définis par Jensen. Les définitions de ces extensions sont données dans ce qui suit:

- Les instances de réseaux: Selon le même principe d'instanciation d'un objet à partir d'une classe dans les langages de programmation objet, les instances de réseaux sont une copie instanciée à partir d'un réseau gabarit (un réseau moule). Notons que différentes instances d'un même réseau peuvent prendre des états différents à un instant donné et sont indépendantes les unes des autres dans tous leurs aspects.
- Le réseau est considéré comme jeton objet : les réseaux de référence implémentent le paradigme des réseaux dans les réseaux. Les places dans ces réseaux (appelés aussi réseaux systèmes) peuvent contenir des jetons schématisant un autre réseau (également appelé réseau objet). Partant de ce principe, on obtient facilement une hiérarchie de

réseaux : un réseau système contenant un jeton schématisant un réseau objet peut lui-même être un jeton schématisant un réseau objet dans un autre réseau système.

- Les canaux synchrones : les canaux synchrones ont été proposés pour la première fois par Christensen et Hansen. Ils permettent de synchroniser deux transitions pour les tirer automatiquement en même temps. Pour que cette synchronisation ait lieu, les deux transitions doivent porter le même nom de canal et le même nombre de paramètres. Notons qu'il est possible de synchroniser plus de deux transitions à un instant donné en les inscrivant, tout simplement, par plusieurs canaux synchrones.
- Les types d'arc : En plus des arcs usuels, les réseaux de référence offrent deux autres types d'arcs : les arcs de réservation et les arcs test. Ces deux types d'arcs sont similaires dans le sens où ils ne changent pas le marquage associé à une place. Les arcs de réservation sont en réalité une autre manière de représenter deux arcs classiques dont les directions sont opposées. Ils permettent de réserver un jeton lors du tirage d'une transition. Les arcs test sont dénués d'orientation et permettent d'accéder (tester) à un jeton donné.

b. Modélisation de la mobilité

Intuitivement, tels les jetons noirs ou les jetons colorés utilisés dans les réseaux de Petri «classiques », les réseaux de référence autorisent une place à contenir un jeton symbolisant un autre réseau de Petri. Voyons cette manière de faire sur les figures 3.1 et 3.2. D'abord, il est inutile de préciser qu'une telle représentation n'est pas très pratique lorsqu'il s'agit de modéliser un système assez complexe. Pour cela, l'outil de modélisation Renew utilise une sorte de pointeur référenciant (d'où le nom : réseau de référence) la fenêtre contenant le réseau jeton. Comme il a été précisé précédemment, dans la terminologie des réseaux de référence, le réseau jeton est appelé réseau objet et le réseau le contenant est appelé réseau système.[34]

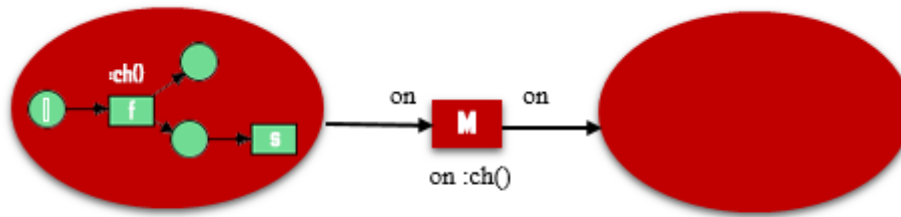


Figure 6. 1: Un réseau objet contenu dans un réseau système

Le réseau objet dans la place de gauche a deux transitions et trois places, la première place possède un jeton ("[]" veut dire jeton) et les deux autres places sont vides. La transition "f" est inscrite avec un canal synchrone ":ch()" (on peut considérer pour faciliter la compréhension que ":ch()" est un autre nom pour la transition "f"). "on" est le nom de ce réseau objet. Le réseau système a une transition et deux places, la première place possède le réseau objet "on" et la deuxième est vide. Le mot « on » écrit sur l'arc de la transition "M" définit le type du jeton à retirer (l'arc en entré) et a mettre (l'arc en sortie), alors le franchissement de "M" provoque l'enlèvement du réseau objet "on" depuis la place gauche et le déposer dans la place droite. La transition "M" est inscrite par le canal "on:ch()". Ça veut dire que, pour franchir "M", il faut que la transition "f" soit validée (la transition "f" inscrit avec le canal ":ch()" est une précondition pour la transition "M"). Dans la figure 3.1 cela est possible et le franchissement synchrone du réseau objet (transition "f") et du réseau système (transition "M") peut-être déclenché. Ceci nous mène à la situation décrite dans la figure 3.2 dans laquelle le réseau objet a été déplacé vers la place de droite de la figure. En même temps et de façon synchrone, le marquage du réseau objet a changé et désormais, aucun franchissement supplémentaire de la transition "f" n'est plus possible.

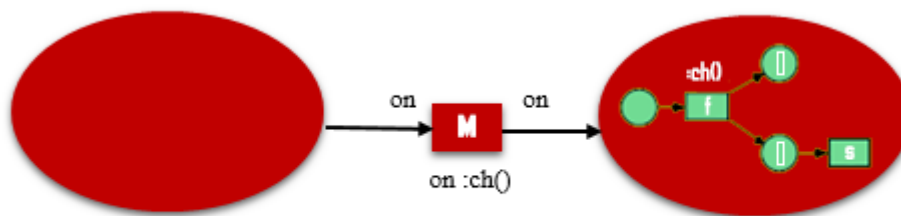


Figure 7. 2: Les réseaux système et objet après le franchissement.

Cet exemple met en avant la manière avec laquelle sont exploitées les interactions entre le réseau objet et le réseau système pour modéliser une entité se déplaçant d'une position vers une autre. En effet, le réseau système peut offrir (ou éventuellement décliner) les possibilités de déplacement autorisant ainsi, à des instants opportuns, les mouvements du réseau objet en activant respectivement la transition inscrite par la pré-condition du canal synchrone (la transition "f") et la transition schématisant le mouvement dans le réseau système (la transition "M"). En l'absence d'un tel point de vue (des jetons vus comme des réseaux), la personne chargée de la modélisation aura à modéliser ou à coder l'entité mobile en faisant appel à des structures de données d'un autre formalisme. Par conséquent, les actions internes de ces entités mobiles vont être négligées lors de cette étape et vont être reléguées à la personne chargée de la modélisation du réseau système enfreignant totalement la réalité. En utilisant le paradigme des réseaux dans les réseaux, on peut aisément harponner la concurrence du système et de l'entité mobile dans un seul et unique modèle sans pour autant désobéir aux obligations d'abstraction. [34]

c. Multi Agent Architecture MULAN

L'architecture de système multi-agents MULAN est basée sur le paradigme des « Nets within Nets », et pour la modélisation via l'utilisation du paradigme des réseaux de référence, nous allons faire appel à l'architecture MULAN pour décrire l'aspect hiérarchique des systèmes multi-agents. Mulan a la structure générale comme représenté sur la figure 3.3. Chaque boîte décrit un niveau d'abstraction en termes de réseau système. Chaque réseau système contient des réseaux d'objets, que la structure est visible par les lignes de Zoom.

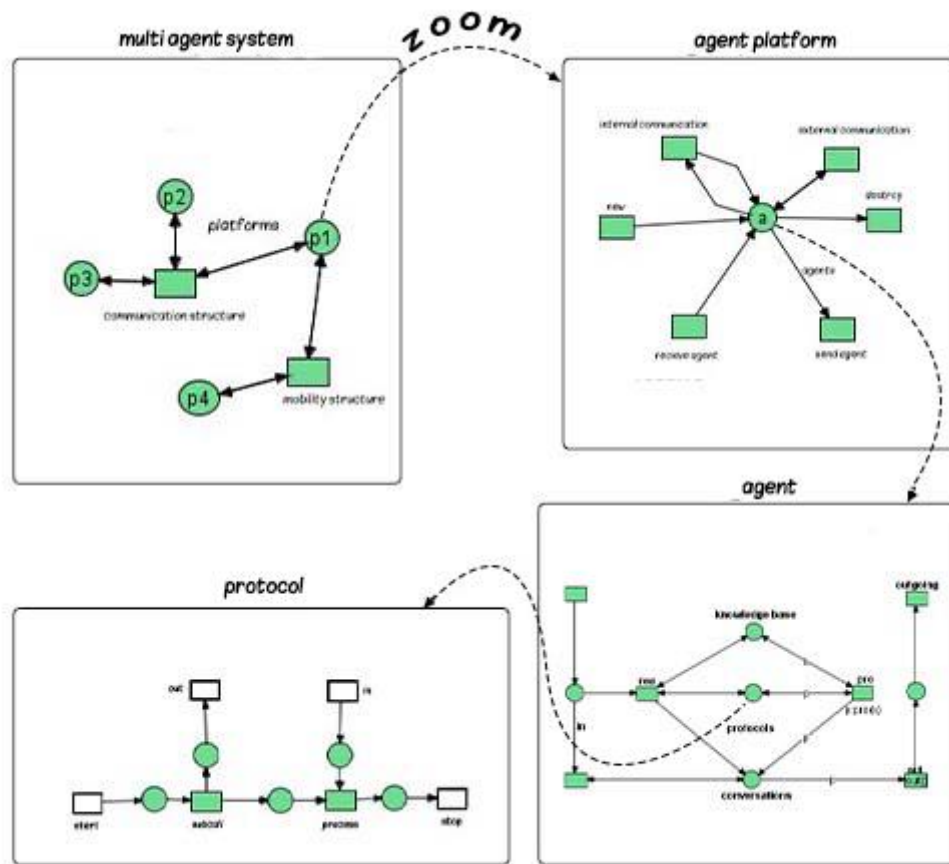


Figure 8.3 : Système multi-agents modélisé avec les réseaux dans les réseaux.

Lors de la modélisation des systèmes complexes, il est fortement déconseillé de vouloir gérer toute la complexité du système au moment de sa modélisation et/ou de son exécution. Ainsi, on introduit la notion de vue d'un système. Plusieurs vues sont possibles : le système multi agents dans sa totalité, les plateformes sur lesquelles les agents sont positionnés, l'agent lui-même ou tout simplement son comportement. Cette notion est prise en compte par Mulan comme le montre la figure 3.3. [35]

4. Outils de modélisation des réseaux de Petri

L'aspect formel des RdP a encouragé les développeurs à mettre au point une multitude d'outils de simulation et de vérification des RdP selon la technique de vérification de modèle (CPNTools, CPNAMI, PROD, JARP, MARIA, LOLA, Petri Net Kernel, GreatSPN, INA, Artifex, ExSpect, FLOWer, f-net, GD ToolKit, Helena, HPSim, INA, JARP, JFern, JPetriNet, Opera, ORIS, PACE, RENEW, ...etc.). La plupart de ces outils présente un environnement graphique d'édition des RdP avec la possibilité de simuler le modèle d'analyser des propriétés génériques des RdP. [36]

Section 2 : Conception (Cas D'étude)

1. Introduction

Avec l'augmentation rapide du nombre de véhicules urbains, la congestion du trafic est un problème sérieux qui concerne la plupart des grandes villes.

Les règles régissant les feux de circulation aux intersections signalisées jouent un rôle crucial dans la régulation et la gestion des systèmes de circulation urbaine, Et pour optimiser les performances de la stratégie de contrôle des feux de circulation, la modélisation, l'analyse et la simulation sur les carrefours routiers urbains sont une grande nécessité.

2. Travaux connexes

Depuis plusieurs années, les travaux sur les VANETs ont été lancés par les institutions universitaires et les laboratoires de recherche industrielle. Bon nombre de recherches ont été consacrées à résoudre le problème de la congestion automobile dans les intersections en utilisant le paradigme multi-agents et les réseaux de petri, qui est la plus appropriée pour cela.

- * Dans [37], les auteurs proposent une nouvelle méthode de contrôle du signal basée sur la rétroaction basée sur le modèle de réseau de Petri. Le procédé effectue une optimisation et une synchronisation en ligne des signaux de trafic. Après avoir terminé le temps minimum du feu vert de la phase actuelle, l'extension du temps du feu vert de la phase actuelle est déterminée en comparant un indice de performance de la phase actuelle avec les phases candidates. L'indice de performance est basé sur le nombre de véhicules en file d'attente et le temps alloué au feu rouge des phases candidates. Enfin, des expériences sur des données de terrain sont menées contre la stratégie de contrôle à heure fixe pour comparer leurs performances sur le retard moyen des véhicules. Les résultats ont vérifié la faisabilité du modèle de réseau de Petri et l'efficacité de la méthode proposée.
- * Un autre travail réalisé [38] parle de l'utilisation de CTM pour développer des stratégies de contrôle des embouteillages induits par des accidents de dispersion et évalue l'efficacité de ces stratégies. Il a également parlé des réseaux de Petri temporisés (TPN)

qui fonctionnent dans les systèmes à événements discrets car ils offrent un équilibre entre la puissance de modélisation et l'analyse. Son travail concentre sur l'utilisation des TPN développés pour modéliser des systèmes de contrôle de feux tricolores variables. De plus, ils proposent également une nouvelle stratégie de contrôle pour éliminer la propagation des embouteillages en empêchant le flux de trafic de se diriger vers un accident.

- * Dans l'article [39] les auteurs montrent les vastes applications des réseaux de Petri dans la modélisation et la simulation, l'analyse et l'évaluation des performances, le contrôle et l'optimisation intelligents et la gestion de la congestion dans les systèmes de trafic urbain. Ce travail décrit notamment les travaux connexes menés à l'aide des réseaux de Petri et discute de sa viabilité, de ses apports et de ses limites. Également, l'extensibilité et le potentiel de recherche future pour approfondir les applications réussies des réseaux de Petri dans les systèmes de trafic sont discutés.
- * Et le travail le plus proche du nôtre est celui d'amal el fallah [40] qui a proposé une architecture adaptative et collaborative pour réguler la congestion du trafic à plusieurs intersections en utilisant le paradigme multi-agent tout en tenant compte des changements dynamiques du flux de trafic. Dans leur travail les agents modélisent les carrefours et peuvent décider de la durée des feux tricolores en fonction de leur perception du flux de circulation. Chaque agent carrefour utilise un arbre de comportement pour mettre à jour l'état des feux tricolores en fonction du nombre de voitures perçues à chaque carrefour. Aussi ils ont démontré également dans leur travail comment les politiques de contrôle dynamique du trafic peuvent être utilisées dans un scénario collaboratif pour réguler le flux de trafic.
- * L'objectif fondamental de notre travail est de fournir une solution pour le problème de congestion du trafic dans les intersections par contrôler les feux de circulation en utilisant le paradigme multi-agents.

3. Description détaillée

Notre environnement est une intersection avec des feux tricolores des deux côtés.

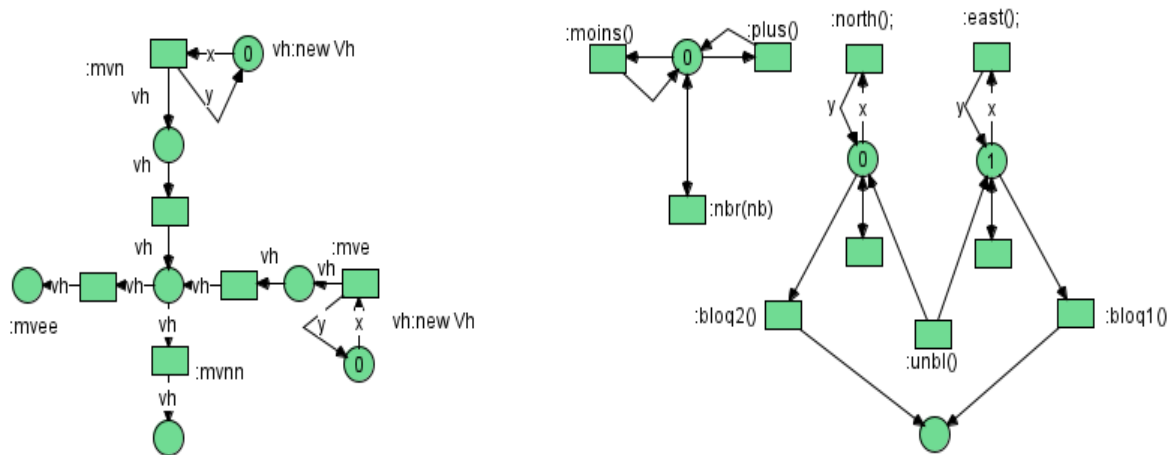


Figure 9.4 : Environnement de système.

Nous avons d'abord à la place située au milieu l'agent d'intersection avec deux routes (Est et Nord), l'agent intersection est responsable de l'intersection afin qu'il ajuste et contrôle le temps de changement des feux de circulation pour réduire le temps d'attente des véhicules et d'éviter les embouteillages inutiles.

Les véhicules sont modélisés comme des agents mobiles où il y a un nombre illimité de véhicules venant des deux directions.

L'environnement est équipé de capteur dont le rôle est de détecter le nombre de voitures dans les deux directions (Est et Nord).

Les feux de circulation prennent deux valeurs (0 ou 1) où 0 indique le feu rouge et 1 indique le feu vert.

Les deux transitions bloq1() et bloq2() sont utilisées pour bloquer les feux lors de la présence d'un véhicule prioritaire (comme si les feux sont éteints), La transition unbl() fait l'opération inverse.

Aussi il y a un compteur qui calcule le nombre de véhicules venant des deux côtés et cela synchronisera avec la transition située dans le protocole de l'agent d'intersection.

Dans notre solution, nous considérons deux types d'agents :

- Agent d'intersection (**Ag**) : Contient les informations (Identifiant d'agent, Nombre de vehicules dans chaque direction, Le mode (équitable, compteur, prioritaire)).
- Agent mobile de véhicules (**Vh**) : Contient les informations (Identifiant d'agent, Direction, Type (ordinaire, prioritaire)).

La structure de ces deux agents est la suivante :

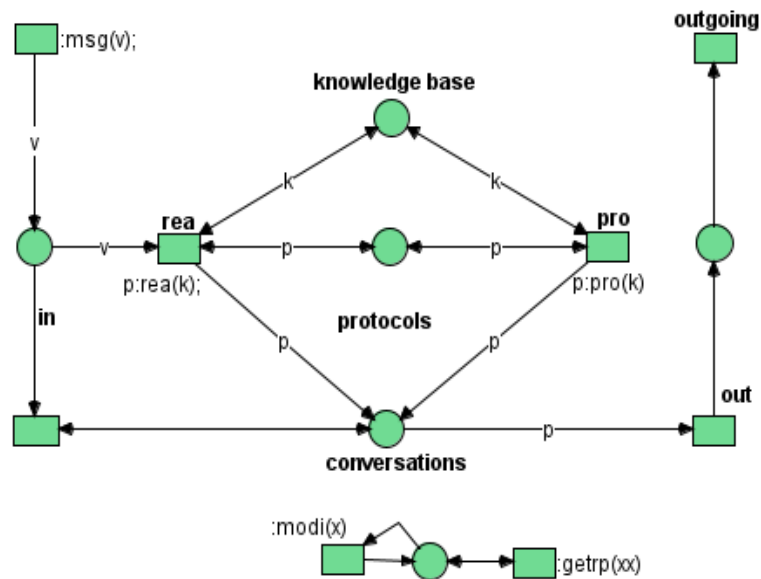


Figure 10.5 : Structure de l'agent d'intersection.

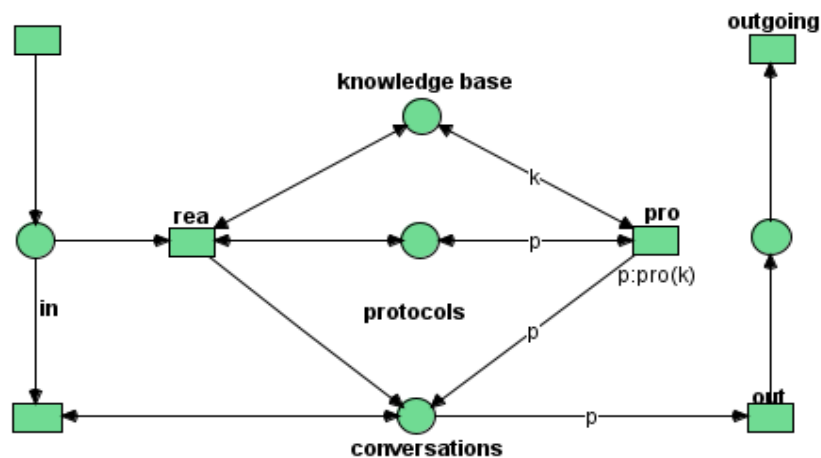


Figure 11.6 : Structure de l'agent de véhicules.

Le protocole de l'agent d'intersection traite trois situation possibles :

Dans le premier cas, les feux de circulation sont modifiés en fonction d'un temp précis afin que les véhicules se déplacent pendant une période de temps et pour la même période de temps. Les transitions east() et north() assurent le changement des feux. Il faut noter que cette solution respecte la section critique. C'est-à-dire qu'à un moment donné seuls les véhicules d'une section (nord ou Est) sont présents dans la place centrale.

Le deuxième cas traite le cas où une section contient un nombre important de véhicules par rapport à l'autre. Dans ce cas l'agent intersection bloquera une section au détriment de l'autre.

Le dernier cas traite la présence d'un véhicule prioritaire dans l'une des deux sections. Dans ce cas l'agent prioritaire bloque la circulation jusqu'au passage de ce véhicule prioritaire. Plus précisément, l'agent test la présence d'un message issu de ce véhicule. Si c'est le cas l'agent bloque les feux : bloq1 et bloq2, Ceci va permettre de bloquer la circulation et seul le véhicule prioritaire peut passer, Ensuite l'agent débloque la situation par la transition unbl().

Nous allons maintenant parler du protocole de véhicule.

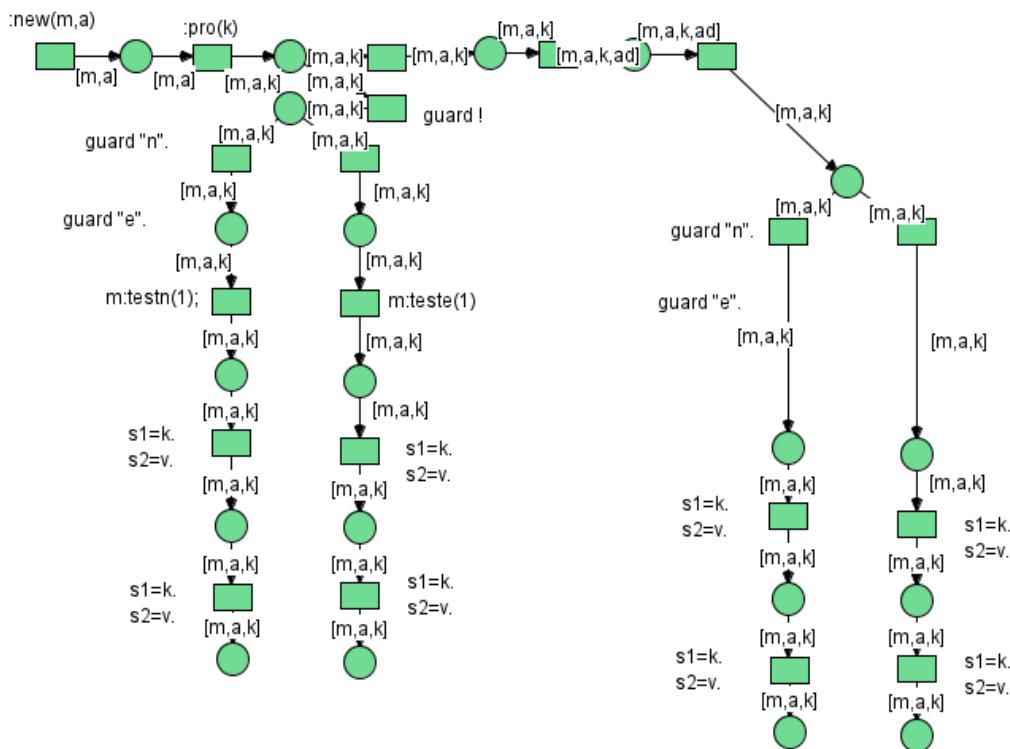


Figure 13.8 : protocole de véhicule.

Lorsque les agents véhicules sont créés, on vérifie d'abord si cette création a généré un véhicule prioritaire (son numéro doit être multiple de 100). Si c'est le cas, il envoie un message à l'agent intersection. Ce dernier bloquera tous les autres véhicules ce qui permettra au véhicule prioritaire de passer en empruntant la section nord ou la section est sans tester les feux rouges et verts. Les autres agents se déplaceront en fonction des feux verts et rouges

La figure 3.9 représente un exemple de simulation du système.

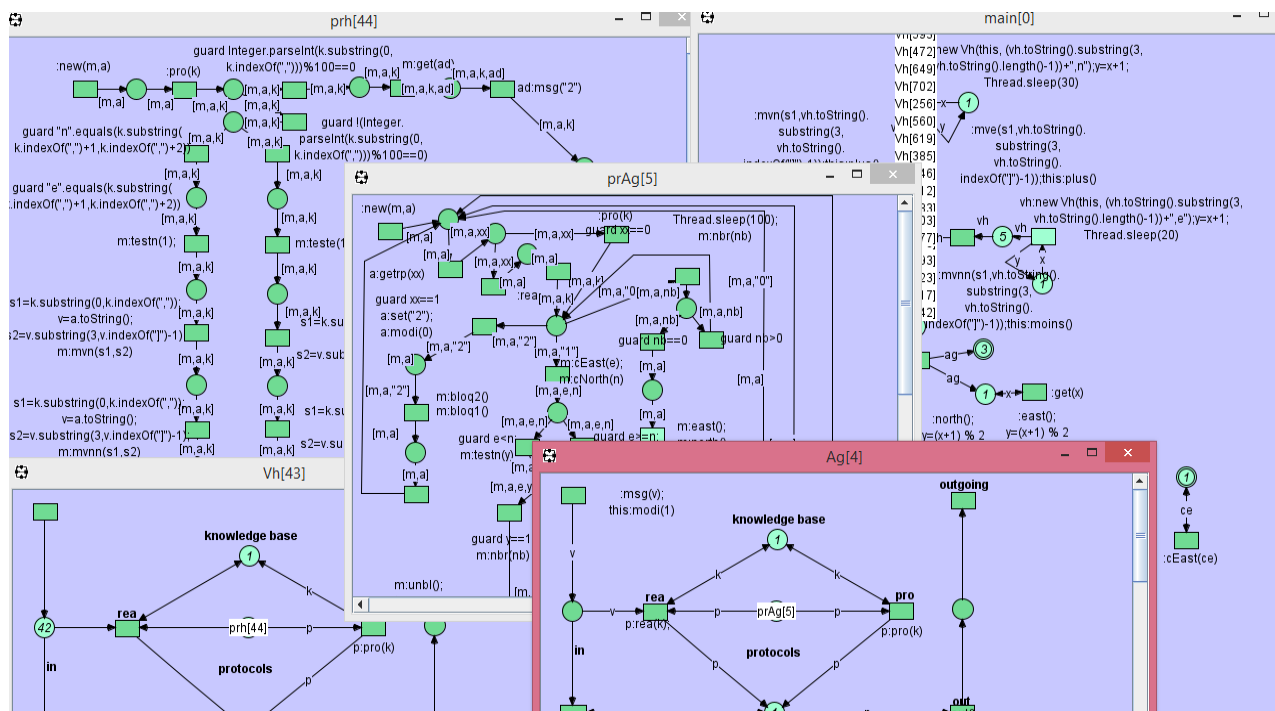


Figure 14.9 : exemple de simulation.

4. Conclusion

Dans ce chapitre, on a essayé de présenter les réseaux de petri en général, ces propriétés, extensions ... par la suite, nous avons étudié les réseaux dans les réseaux, les réseaux de référence, l'architecture des systèmes multi-agents et nous avons étudié certains aspects de notre problème et proposé une solution.

L'objet du prochain chapitre sera l'implémentation de notre système.

Chapitre IV Implémentation

1. Introduction

Après avoir réalisé notre problème et fait la conception de notre système proposé et vu la solution proposée antérieurement qui vise à contrôler les feux de circulation dans les intersections pour réduire la congestion routière dans le chapitre précédent. on va maintenant entamer à l'étape d'implémentation.

Nous exposerons les outils et langages utilisés durant le processus de développement, et nous expliquerons brièvement les différentes fonctionnalités de celui-ci.

2. Outils de conception

- *Renew (The Reference NetWorkshop)*

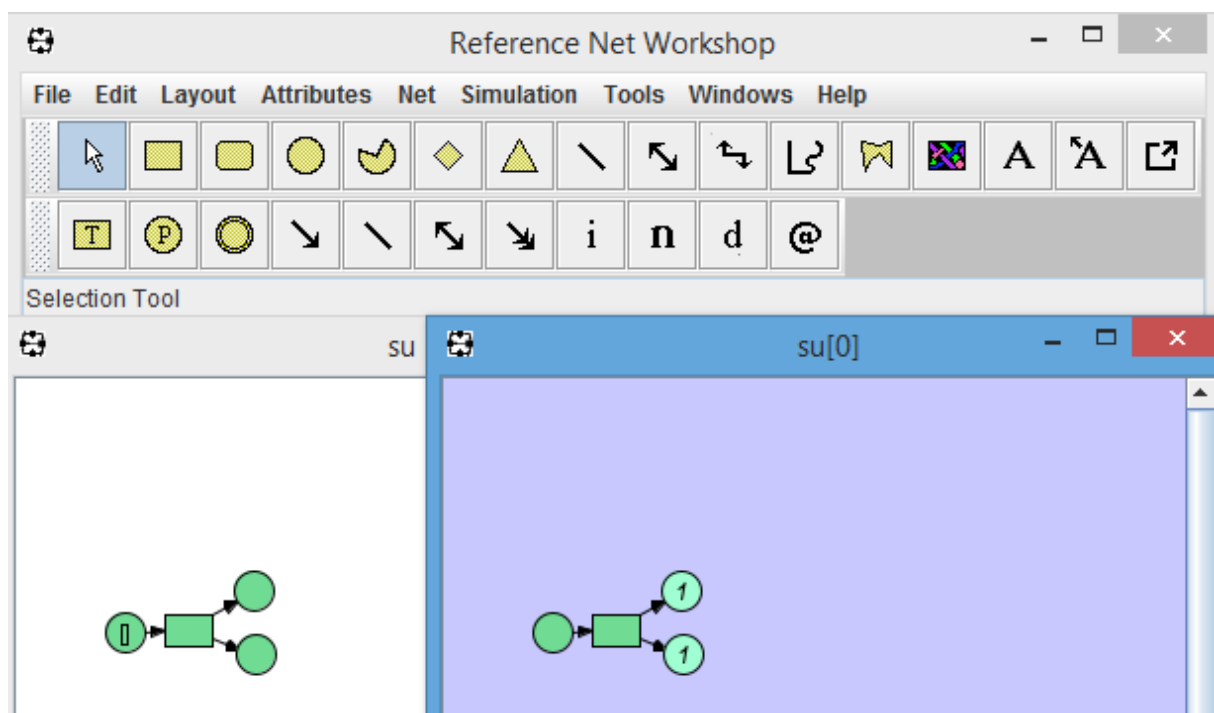


Figure 4.15 : *Renew GUI, Petri net.*

La principale force de Renew réside dans son ouverture et sa polyvalence. [42]

- * Renew a été écrit en Java, il fonctionnera donc sur tous les principaux systèmes d'exploitation modernes sans modifications.
- * Renew est livré complet avec la source, de sorte que ses algorithmes peuvent être librement étendus et améliorés.

- * Il est en effet possible d'ajouter rapidement des inscriptions spéciales filet. Il est même possible d'implémenter des formalismes nets complètement nouveaux sans changer la structure de base de Renew.
- * Renew peut utiliser n'importe quelle classe Java. Aujourd'hui, il existe des classes Java qui couvrent presque tous les aspects de la programmation.
- * Les réseaux de référence sont eux-mêmes des objets Java. Passer des appels de code Java vers des réseaux est aussi simple que de passer des appels de réseaux vers du code Java.
- * Le formalisme du réseau de Petri de Renew pourrait également être très intéressant pour les développeurs.
- * Renouveler prend en charge les canaux synchrones. Les canaux sont un mécanisme de communication puissant et peuvent être utilisés comme un concept d'abstraction fiable.
- * Les instances de réseau permettent une modélisation orientée objet avec des réseaux de Petri. Alors que quelques autres formalismes nets fournissent des instances nettes, c'est leur intégration cohérente avec les autres fonctionnalités qui les rend utiles.
- * Les réseaux de référence ont été spécifiquement conçus avec le ramasse-miettes des instances de réseau à l'esprit, ce qui est indispensable pour une bonne programmation orientée objet.
- * De nombreux types d'arcs sont disponibles et couvrent presque tous les formalismes nets. Le temps de simulation avec une sémantique de temps de déclenchement au plus tôt est intégré.

Il y a cependant quelques points à prendre en compte.

- * Il n'existe actuellement que des outils d'analyse rudimentaires pour Renew. Bien que quelques interfaces d'exportation aient déjà été implémentées, une analyse utile semble encore loin. Actuellement, Renew s'appuie entièrement sur la simulation pour explorer les propriétés d'un réseau, où vous pouvez explorer de manière dynamique et interactive l'état de la simulation.

Cependant, pour de nombreuses applications, l'analyse ne joue pas un rôle prédominant. Les réseaux de Pétri ne sont souvent utilisés qu'en raison de leur représentation graphique intuitive, de leur expressivité et de leur sémantique précise.

- * Pendant la simulation, l'utilisateur ne peut changer le marquage actuel du filet simulé qu'en déclenchant une transition. Cela peut rendre un peu plus difficile la configuration d'un scénario de test souhaité.

- * Dans notre formalisme, il n'y a pas de notion de probabilités ou de priorités de tir. En exploitant l'architecture ouverte de Renew, ces fonctionnalités peuvent être ajoutées ultérieurement, éventuellement en tant que contributions de tiers.
- * Renew est un outil académique.

3. Langage de programmation utilisée

Le langage de programmation utilisée pour développer le système est : Java. [43] C'est un langage de programmation orienté objet développé par Sun Microsystems, Java est à la fois un Langage de programmation et un environnement d'exécution. Il est très performant et très utilisé par un grand nombre de développeurs grâce à plusieurs caractéristiques parmi les quelle on cite :

- **Portabilité** : Permet de crée des applications compatible avec des nombreux systèmes d'exploitation grâce à JVM « java virtuelle machine », java donne aussi la possibilité de développer des applications sur mobile, desktop et sur le web.
- **Sécurisé** : La sécurité fait partie intégrante du système d'exécution et du compilateur. Un programme Java planté ne menace pas le système d'exploitation.
- **Libre** : java est open source et avec pleine de documentations, ainsi que la JVM est gratuite.

Le développement de notre système nécessite l'utilisation de l'aspect desktop.

L'aspect desktop de Java nécessite l'installation d'un Kit de Développement Java (JDK). Le JDK est un ensemble de bibliothèques de base et d'outils avec les quels le code source java peut être compilé et transformé en byte-code afin que la JVM (Java Virtual Machine) puisse l'interpréter.

4. Plateforme utilisée

- *Jade*

JADE (Java Agent Development Framework) [44] est une plateforme multi agents écrite en JAVA, développé par Telecom Italia. JADE est composée d'un ensemble d'outils nécessaire au développement et à la mise en service des systèmes multi agents au sein d'un environnement spécifique. Les SMA développés dans la plateforme JADE sont conformes aux normes FIPA (ensemble de standards qui permettent d'interpréter les communications entre agents de manière respecter leur sens initial).

En utilise JADE pour les raisons suivantes :

- * Il est simple de créerdes agents avec jade.
- * Jade gère la communication entre les agents et offre des interfaces de gestion des agents.
- * Le mode de communication direct (avec des messages) est supporté par la plate-forme JADE en utilisant lelangage FIPA ACL.
- * La plate-forme Jade répond à plusieurs fonctionnalités et offre une large gamme de bibliothèques.
- * Les agents développés en JADE sont écrits totalement en JAVA qui est un langage facile et basé sur la notion d'objet.

JADE fournit :

- Un environnement où les agents JADE sont exécutés
- Bibliothèques de classes pour créer des agents utilisant l'héritage et la redéfinition des comportements
- Une boîte à outils graphique pour surveiller et gérer la plateforme d'agents intelligents

JADE possède trois modules principaux (nécessaires aux normes FIPA) :

- DF « Director Facilitator » fournit un service de « pages jaunes » à la plate-forme.
- ACC « Agent Communication Channel » gère la communication entre les agents.
- AMS « Agent Management System » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

Ces trois modules sont activés à chaque démarrage de la plateforme. La figure 4.2 montre l'interface graphique de JADE :

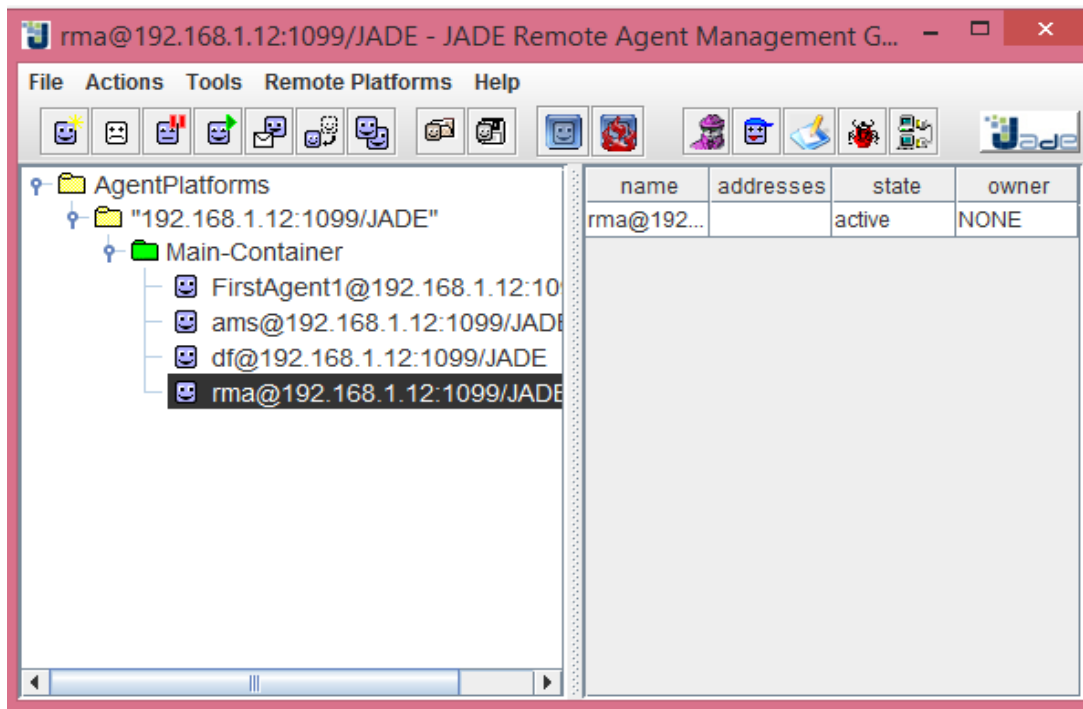


Figure 16.2: Interface graphique de jade

5. Environnement de développement

- **IntelliJ IDEA**

IntelliJ IDEA [45] est un IDE intelligent et tenant compte du contexte qui permet de travailler sur toutes sortes d'applications en Java et dans d'autres langages de la JVM tels que Kotlin, Scala et Groovy. De plus, IntelliJ IDEA Ultimate aide à développer des applications web full-stack grâce à ses puissants outils intégrés, à la prise en charge de JavaScript et de ses technologies connexes et à la prise en charge avancée de frameworks populaires tels que Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus et Helidon. IntelliJ IDEA peut être complété par des plugins gratuits développés par JetBrains afin de pouvoir travailler avec d'autres langages de programmation, parmi lesquels Go, Python, SQL, Ruby et PHP.

Certaines de ses caractéristiques :

- Démarrage facile
- Des raccourcis clavier pour tout
- Thèmes standard et personnalisés
- Accessibilité

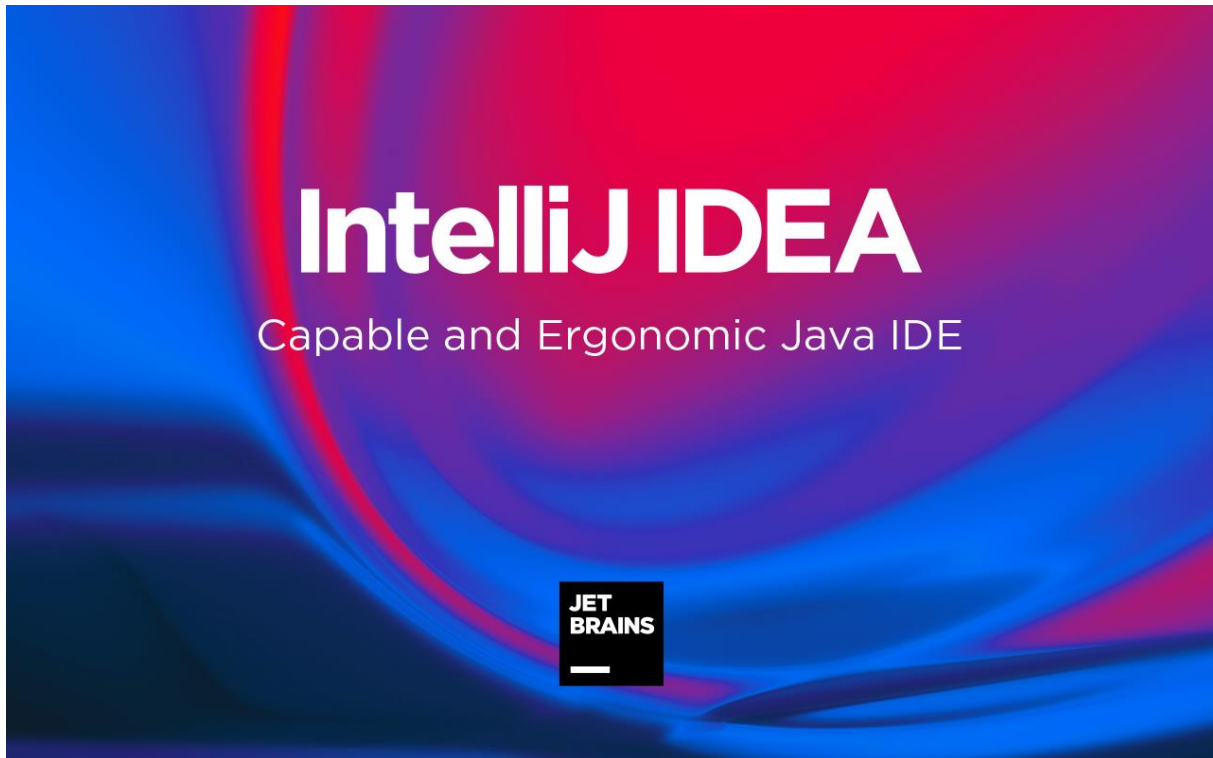


Figure 17.3: *IntelliJ IDEA IDE*

6. Interface

Nous avons fait un système multi agents qui applique de manière simplifiée ce que nous avons expliqué précédemment dans le chapitre de conception, Le rôle principal de système est d'essayer de réduire les accidents de la circulation dans une intersection en utilisant des feux de circulation, et pour y parvenir, nous avons besoin d'une intersection, feux de circulation et de voitures comme indiqué dans la figure ci-dessous.

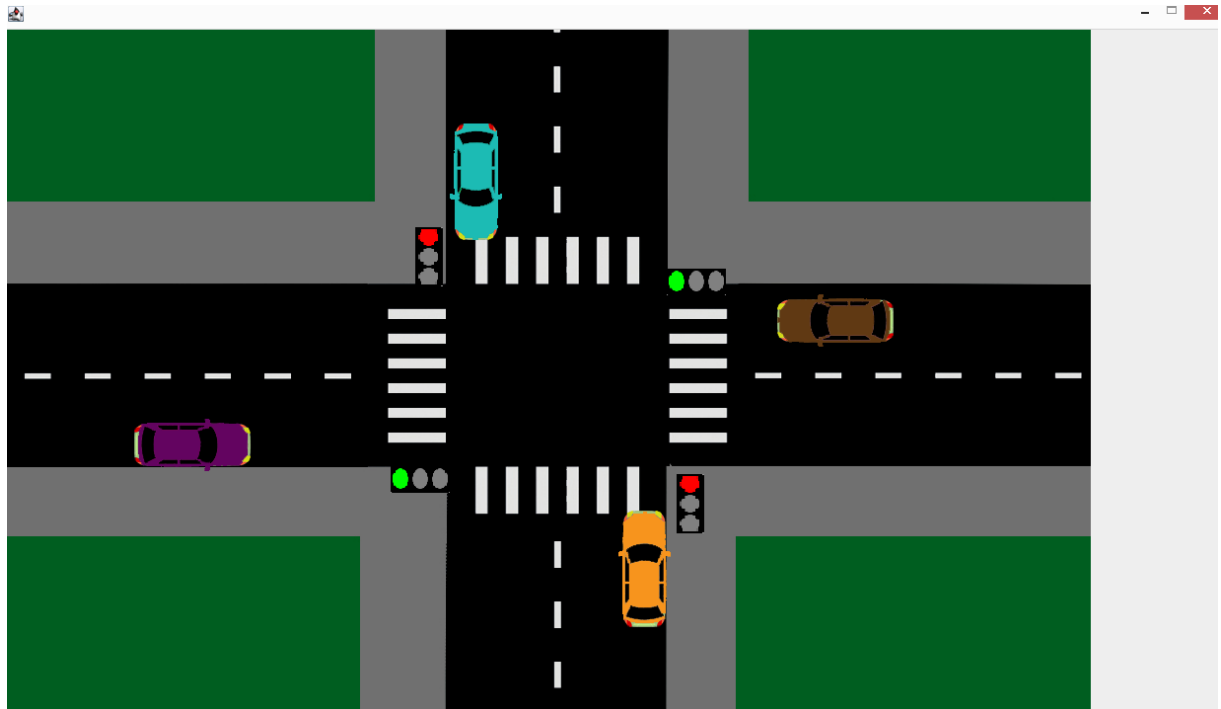


Figure 18.4: Interface de logiciel.

Nous pouvons voir plus de détails sur le fonctionnement du système en utilisant la fenêtre dans la figure 4.5 qui représente l'agent "Sniffer" qui est un outil de JADE permet d'intercepter les messages ACL pendant leur changement et les montrer graphiquement utilisant une notation semblable à UML.

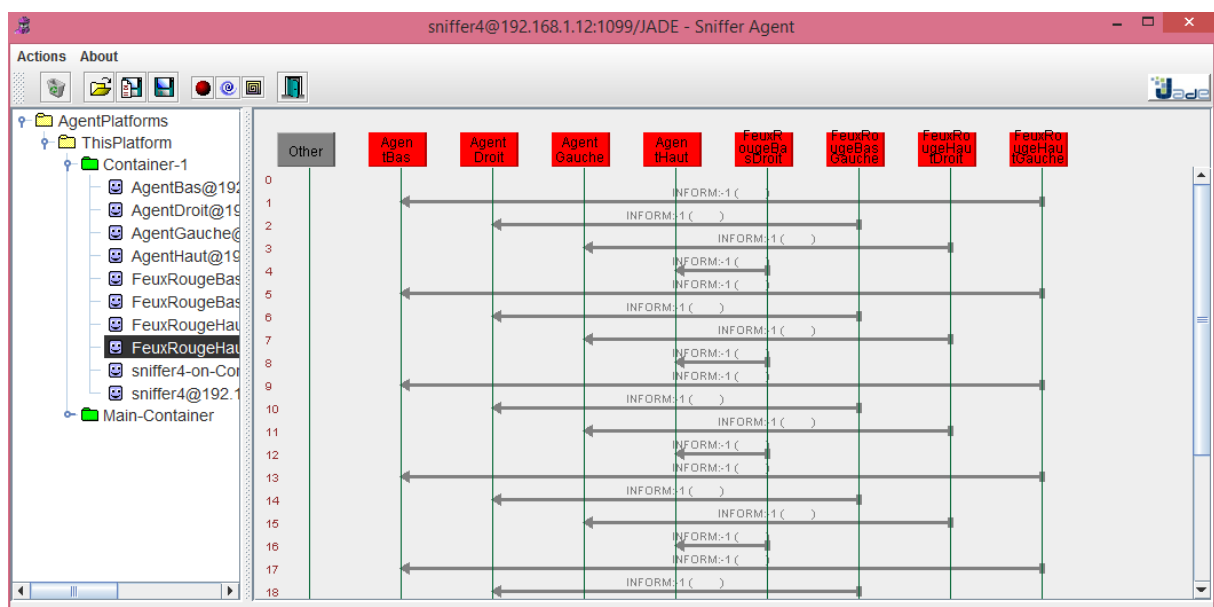


Figure 19.5: fenêtre de l'agent "Sniffer".

Il y a quatre feux de circulation et quatre voitures et ils sont tous considérés comme des agents, Le contrôleur principal ici est les feux de circulation afin qu'ils changent les couleurs verte et rouge selon une période spécifique que nous fixons comme nous le voulons, puis envoient un message aux voitures voisines, permettant soit de continuer s'il est vert ou arrête si c'est rouge.

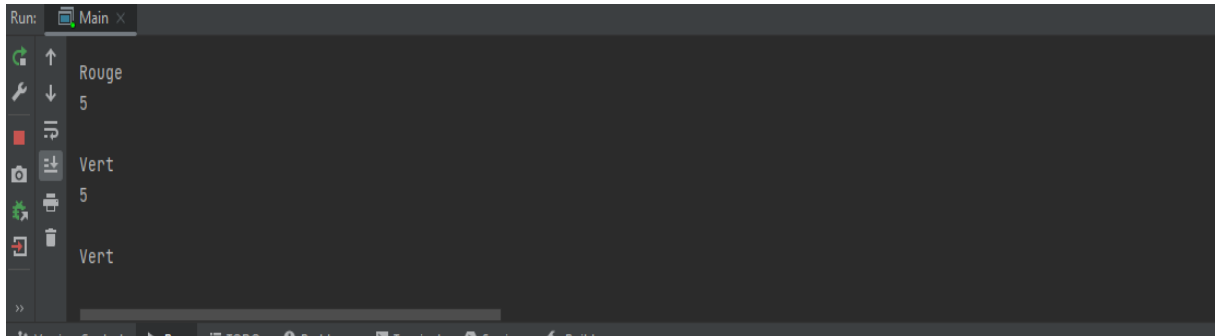


Figure 20.6: *message d'exécution.*

Lorsque en exécute le programme, ce message nous montre combien de fois les couleurs des feux de circulation a changé.

7. Conclusion

Dans ce chapitre nous avons introduit l'environnement et les langages de développement utilisée, aussi faire une aperçu sur notre système multi agents. Dans celui-ci, nous avons essayé d'appliquer ce que nous avons fait dans le chapitre de la conception avec l'outil renew d'une manière plus simplifiée

Conclusion générale

Conclusion générale

Trouver une solution aux problèmes de circulation, qu'il s'agisse d'accidents ou de longues périodes d'attente, est devenu une chose très importante dans la vie des gens, car c'est devenu l'ère de la vitesse à tous égards, de sorte que la plupart de la population transport par des voitures pour se rendre au travail, études, voyages et autres. Par conséquent, il est devenu nécessaire de trouver des solutions qui réduisent les risques et facilitent la vie, et ici dans notre projet, nous avons proposé une solution pour réduire le temps d'attente et réduire les accidents de la circulation tout en donnant la priorité aux ambulances, à la police ou aux piétons aux intersections en utilisant les VANETs qui sont récemment devenues populaires dans les recherches nouvelles et futures.

Dans ce travail nous avons mentionné les réseaux de Petri qui sont utilisées spécifiquement comme la chose la plus utile pour faire la simulation, aussi nous avons utilisé l'outil RENEW car c'est un outil robuste et facile à manipuler, permet de réduire considérablement l'important effort initial en termes d'heures-homme nécessaires à la construction et la validation du modèle de test. De plus, la structuration en quatre couches (multi-agents, plateforme, agent et protocoles comportementaux) du système selon l'architecture Mulan, rend aisé le processus de modélisation, de validation du modèle et de génération des cas de tests en raison de la taille réduite des parties considérées. L'approche proposée s'appuie fortement sur les résultats de la simulation produits par cet outil.

Notre objectif a été de mise en place d'une simulation du système qui contrôle les feux de circulation ainsi que la gestion du trafic urbain. Dans le chapitre Conception, nous avons expliqué le principe que nous faisons pour que nous mentionnions tous les agents et le travail de chacun d'eux, ainsi que le fonctionnement des protocoles.

Grace à la plateforme utilisée : JADE, nous avons pu simuler un système des feux de circulation intelligents basé sur les systèmes multi-agents pour la régulation du trafic routier au niveau des intersections. Le rôle principal de système est de réduire les problèmes de congestion.

Le sujet dans lequel nous nous sommes lancés est très vaste et le travail réalisé peut être amélioré et suivi afin d'en faire un système plus dynamique et autonome. Parmi les perspectives à prendre en compte pour améliorer le fonctionnement du système, une amélioration des feux pour détecter l'intention des piétons qui traverse la route et anticipe leur demande éventuelle

pour que le feu passe au vert directement. Développer et améliorer la communication des véhicules entre eux et connaître la direction de chacun d'eux pour éviter de tomber dans des accidents potentiels, et ce travail peut également être élargi de manière significative en communiquant les intersections adjacentes entre eux et en envoyant des messages sur la présence des Erreurs.

Bibliographie

- [1] Fleisch, E. (2013) What is the Internet of things? [cited 2013 May 20]; available from <http://www.im.ethz.ch/education/HS10/AUTOIDLABS-WP-BIZAPP-53.pdf>
- [2] [https://doi.org/10.1016/S1570-8705\(03\)00043-X](https://doi.org/10.1016/S1570-8705(03)00043-X)
- [3] <https://doi.org/10.1145/1160633.1160743>
- [4] Contribution au système de transport intelligent : sécurité des communications dans les réseaux Ad hoc véhiculaires par Youssef Inedjaren
- [5] Journal of Network and Computer Applications Volume 37, January 2014
- [6] A. Bachir, A. Benslimane, «A multicast protocol in ad hoc networks inter-vehicle geocast», Vehicular Technology Conference, 2003. The 57th IEEE Semiannual, Volume: 4, On page(s): 2456- 2460.
- [7] Saif Al-Sultan n, Moath M. Al-Doori, Ali H. Al-Bayatti, Hussien Zedan. « A comprehensive survey on vehicular Ad Hoc network», Journal of Network and Computer Applications. Article history Received 20 June 2012 Received in revised form 21 February 2013.
- [8] M. Burmester, E. Magkos, and V. Chrissikopoulos. « Strengthening Privacy Protection in VANETs », in IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB), Avignon, 2008.
- [9] Q. Xu and D. Jiang, « Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum », Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual, vol. 4, pp. 2451-2455, Apr. 2003.
- [10] M. JERBI, « Protocoles pour les Communications dans les Réseaux de Véhicules en environnement Urbain : Routage et GeoCast basés sur les intersections », Université D'EVERY VAL D'ESSONNE, thèse de doctorat, 2008.
- [11] CHAIB, Noureddine (2011) La sécurité des communications dans les réseaux VANET. Magister thesis, Université de Batna 2.
- [12] DOI: 10.1109/ITST.2006.289012

- [13] Moez JERBI, "Protocoles pour les communications dans les réseaux de véhicules en environnement urbain: Routage et GeoCast basés sur les intersections", Thèse de Doctorat, Université d'Evry val d'Essonne, 06 novembre 2008.
- [14] Mostafa M. I. Taha B.Sc. « Electrical Engineering, Broadcasting Protocols in Vehicular Ad-Hoc Networks (VANETs) », Assiut University, 2004.
- [15] Ait Ali Kahina. « Modélisation et Etude de Performances dans Les Réseaux VANET ». Thèse de doctorat de l'Université de Technologie de Belfort-Montbéliard. 2012.
- [16] Tayeb Lemlouma, « Le routage dans les réseaux mobiles ad hoc »
[www.opera.fr/people/TayebLemlouma/papers/AdHoc_presentation .PDF](http://www.opera.fr/people/TayebLemlouma/papers/AdHoc_presentation.PDF)
- [17] Amina Naimi-Meraihi, « Délai de routage dans les réseaux Ad Hoc 802.1 », Thèse doctorat. INRIA Rocquencourt, 2005.
- [18] M. Gerla, X. Hong, G. Pei, « Fisheye State Routing Protocol (FSR) », IETF Internet Draft, work in progress, draft ietfmanet- fsr-03.txt, July 2002.
- [19] T. Clausen et al. « Optimized Link State Routing Protocol (OLSR) », RFC 3626, Network Working Group, 2003.
- [20] Catherine Loison, Thomas Ruocco, Camille Rives, « Routage Multicast dans les Réseaux Véhiculaires (VANET) ». Université d'Avignon, 2013.
- [21] Z. J. Haas, « The Zone Routing Protocol (ZRP) for ad hoc networks », Internet Draft, 1997.
- [22] Navid Nikaein, Christian Bonnet and Neda Nikaein, « HARP - Hybrid Ad Hoc Routing Protocol », in proceeding of IST 2001: International Symposium on Telecommunications, Iran/Tehran 2001.
- [23] AOUES Abdelaziz, HAMMOUDI Meryam, BENAÏSSA Yousef, BENSALDANE Nadjat. « Les Réseaux Véhiculaires VANET », Département d'informatique L3 GTR Technologie Réseau 2014/2015, Université des sciences et de technologie HOUARI BOUMEDIENE.
- [24] Yang Xia, Chai Kiat Yeo, Bu Sung Lee, « Hierarchical Cluster Based Routing for Highly Mobile Heterogeneous MANET », Communications and Information Technologies, ISCIT'07, International Symposium on, pp. 936–941, Oct. 2007.
- [25] C. A. Petri. "Kommunikation mit Automaten". Bonn : Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962 (en allemand) aussi dans : New York : Griffiss Air Force Base, Technical Report RADC-TR-65--377, Vol. 1, pp. Suppl. 1.

- [26] TADA0 MURATA "Petri Nets: Properties, Analysis and Applications" Fellow, IEEE, Invited Paper. Proceedings of IEEE, vol. 77, NO 4, April 1989.
- [27] Alexandre Pauchet, "Modélisation des Systèmes Complexes Introduction aux Réseaux de Petri" 800A-2, pauchet@insa-rouen.fr INSA Rouen - Département ASI.
- [28] G. Scorletti et G. Binet Réseaux de Petri Maîtres de conférences à l'Université de Caen, France 20 juin 2006. Page web: http://www.greyc.ensicaen.fr/EquipeAuto/GerardS/mait_Petri.html.
- [29] S. LAFTIT, J.M. PROTH and X.L. XIE, "Performance Analysis of Stochastic Jop-Shop Systems Using Event Graphs", ECC 91, Grenoble, France, July 2-5 1991, pp. 1484-1489.
- [30] Wael KHANSA, "RESEAUX DE PETRI P-TEMPORELS CONTRIBUTION A L'ETUDE DES SYSTEMES A EVENEMENTS DISCRETS".
- [31] Jensen Kurt. Coloured petri nets: Basic concepts, analysis methods and practical use, volume 1.
- [32] Modelling Mobility and Mobile Agents Using Nets within Nets
- [33] Kummer O, Simulating synchronous channels and net instances, in J. Desel, P. Kemper, E. Kindler, and A. Oberweis (Eds.), Forschungsbericht Nr. 694: 5. Workshop Algorithmen und Werkzeuge für Petrinetze, pp. 73–78. University of Hamburg, ComputerScience Department, 1998.
- [34] Andrea Asperti and Nadia Busi. Mobile Petri nets. Technical report, Department of computer science, University of Bologna, TR UBLCS-96-10, 1996.
- [35] Bernhard Bauer, James Odell, and H. van Dyke Parunak. Extending UML for Agents. In Proceeding of Agent-Oriented Information Systems Workshop, pages 3–17, 2000.
- [36] http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/complete_db.html
- [37] A New Traffic Signal Control Method Based on Hybrid Colored Petri Net in Isolated Intersections
- [38] Design of Variable Traffic Light Control Systems for Preventing Two-Way Grid Network Traffic Jams Using Timed Petri Nets Jiliang Luo , Senior Member, IEEE, Yi-Sheng Huang , Senior Member, IEEE, and Yi-Shun Weng

[39] A Review on the Applications of Petri Nets in Modeling, Analysis, and Control of Urban Traffic Kok Mun Ng, Member, IEEE, Mamun Bin Ibne Reaz, Member, IEEE, and Mohd Alauddin Mohd Ali, Member, IEEE

[40] Smart Mobility Using Multi-Agent System Assia Belbachira,b, Amal El Fallah-Seghrouchnib, Arthur Casals,a,b,d, Marcia Pasinc

[41] Michael Wooldridge. Intelligent Agents, chapter 1. The MIT Press, 1999.

[42] Renew – User Guide

[43] <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-java-programming/>

[44] <https://jade.tilab.com/>

[45] <https://www.jetbrains.com/fr-fr/idea/>