



# Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil

---

Sébastien Faye

**2012D002**

Mars 2012

Département Informatique et Réseaux  
Groupe RMS : Réseaux, Mobilité et Services

# Rapport Technique

## Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil

Sébastien Faye

Sous la direction de Claude Chaudet et Isabelle Demeure

Institut Télécom, Télécom ParisTech CNRS LTCI UMR 5141 ; 46 rue Barrault, 75013 Paris

{*prenom.nom@telecom-paristech.fr*}

Février 2011, deuxième version

### Résumé

Le trafic routier urbain est au cœur de nombreuses problématiques : plus encore ces dernières années, cet aspect critique intervenant au quotidien est défavorable à de nombreux domaines, tels que l'économie ou encore l'écologie. Pour ces raisons, les systèmes de transport intelligents (STI) sont apparus depuis la fin des années 1990, permettant d'optimiser au mieux les dépenses et l'expérience de l'utilisateur sur des réseaux souvent complexes. Dans cet article, après avoir étudié les horizons de tels systèmes, nous concentrerons une grosse partie de notre état de l'art sur une technologie dynamique et utilisée notamment dans les systèmes distribués : les réseaux fixes de capteurs sans fil. Nous verrons alors que ces dispositifs peuvent se révéler souples dans le cadre des STI, et participer à faible coût à l'obtention de résultats intéressants.

**Mots clés:** Systèmes de transport intelligents, Gestion du trafic routier, Réseaux de capteurs sans fil.

## 1 Introduction

De nos jours, se déplacer est devenu un aspect essentiel de la vie quotidienne : qu'il s'agisse de transports en commun ou de véhicules personnels, le vaste réseau formé de ces moyens de locomotion est immensément complexe à gérer. Sa gestion recouvre l'ensemble des techniques humaines et automatisées permettant d'assurer la surveillance des transports, au mieux un gain de performance dans l'acheminement des différents flux. Dans ce document, nous nous attardons sur le cas du trafic routier urbain et effectuons un état de l'art de son contrôle par un réseau fixe de capteurs sans fil. Notons que le cas de réseaux mobiles ne sera que faiblement abordé car représente un aspect particulier, sélectif et parfois utopique des systèmes distribués.

L'objectif principal de ce document est de fournir un support récapitulatif d'un domaine dispersé dans de multiples thématiques de la littérature, et de l'allier à l'aspect novateur et dynamique que représentent les systèmes distribués. Parallèlement, notre centre d'intérêt est de présenter le cas particulier de la gestion des feux de circulation en milieu urbain, ceci en s'appuyant sur la littérature utilisant les réseaux fixes de capteurs sans fil et en présentant des outils permettant de simuler le contrôle du trafic routier urbain.

Ce document se décompose comme suit : dans l'immédiat, nous introduisons quelques définitions autour des types de réseaux existant ainsi que leurs métriques et principaux paramètres ; dans la partie suivante, nous abordons les systèmes de transports intelligents sous un aspect général, en faisant un rapide tour sur l'existant et les principes directeurs ; dans la deuxième partie, nous parlons plus en détail de l'intérêt de l'utilisation de réseaux de capteurs sans fil pour contrôler le trafic routier urbain, et

établissons un état de l'art ; enfin, nous présentons des outils de simulation permettant de modéliser le trafic routier et concluons.

## 1.1 Définitions

Nous définissons ici des termes généraux essentiels à la compréhension de ce document. Ces derniers serviront par la suite à caractériser les types de réseaux existant.

**Définition 1. Les systèmes de transport intelligents (STI)** apparaissent comme étant "*l'application des technologies de l'information et de la communication au domaine des transports*" ([21]). Le terme système est vague et se décline en un ensemble de moyens mis en place pour gérer au mieux les contraintes liées au trafic routier, telles que les embouteillages, la sécurité ou même la pollution. En particulier, ces systèmes offrent un caractère réactif à une infrastructure pourtant fixe, mais dont la population peut grandement varier.

**Définition 2. Les réseaux de capteurs sans fil** appartiennent à la famille des réseaux mobiles ad hoc (MANET) et se composent d'un large ensemble de capteurs à capacité et énergie généralement limitées. Dans de nombreux cas, les capteurs sont constitués des unités suivantes ([58]) :

- **Unité d'acquisition**, permet le recueil de données environnementales et la conversion analogique vers numérique.
- **Unité de calcul**, permettant le lancement de procédures, protocoles et autres.
- **Unité de communication**, permettant la connexion au réseau (émission et réception). Ceci, en sans-fil (ex : radio), souvent en multi-sauts et permettant de s'affranchir des inconvénients filaires (temps d'installation et facilité d'accès). La logique est ici distribuée.
- **Unité d'énergie**, qui permet la répartition de l'alimentation entre les différents constituants. Dans de nombreux cas, les capteurs sont dispersés dans des zones pauvres en énergie, et sont dotés d'une batterie non-rechargeable et non-renouvelable ([27]). Dans le cas du trafic routier, il est néanmoins possible voire nécessaire de s'abstenir de contrainte énergétique.

Le rôle de chaque capteur est de récolter un ensemble de données dans son environnement, et le transmettre de proche en proche jusqu'à atteindre généralement une station de base, qui peut jouer le rôle de coordinatrice du réseau et communiquer vers l'extérieur les données importantes recueillies. L'utilisation de tels réseaux est répandue dans de nombreux domaines et applications : nous pouvons par exemple citer la surveillance de forêts, d'infrastructures critiques, ou encore la détection d'agents biochimiques dans le secteur militaire.

## 1.2 Métriques et paramètres

L'application des technologies de l'information et de la communication au domaine des transports peut être vu de différentes manières. La plupart de ces dernières sont dites réactives car permettent de recueillir des données propres à la circulation routière afin de la surveiller ou même tenter de la contrôler. Globalement, leur utilisation requière donc la connaissance de certaines métriques propre à l'environnement urbain dans lequel elles se situent. A une échelle plus locale, les technologies sont utilisées à des endroits stratégiques du réseau, comme les intersections dont nous détaillons le fonctionnement classique dans la partie suivante.

**Définition 3. Un capteur électromagnétique** est un type de capteur sans fil fixe utilisé notamment pour la gestion du trafic routier. L'unité d'acquisition utilisée est un magnétomètre, permettant d'enregistrer les variations du champ magnétique terrestre. Comme indiqué ci-dessus, la contrainte énergétique peut facilement être levée, mais certains auteurs considèrent toutefois l'utilisation de batteries en conseillant leur durée à 2 ou 3 ans ([43]). Dans le cadre des STI, le rôle de tels dispositifs va être de relever des informations sur le trafic routier.

**Définition 4.** Plus globalement, un **détecteur** est une technologie permettant de récolter et transmettre des informations à un nœud traitant. Ce terme englobe donc les capteurs électromagnétiques, mais également les autres technologies existantes et détaillées plus loin dans le document.

Ceci nous permet de définir deux types de réseaux liés au trafic routier urbain.

**Définition 5.** Un **réseau fixe** regroupe un ensemble de technologies (dont détecteurs) étant fixes sur le terrain : leur position n'est pas amenée à être changée, mis à part pendant des périodes d'entraînement où l'objectif va être de déterminer leur emplacement définitif. Dans ce document, nous étudions essentiellement ce type de réseau.

**Définition 6.** Un **réseau véhiculaire** est un cas des réseaux de capteurs sans fil où les capteurs sont mobiles. Par ce terme, nous désignons des réseaux de véhicules où chaque acteur possède un capteur embarqué, ceci rendant possible la mobilité.

**Définition 7.** Un **système coopératif** est un réseau hybride où des acteurs mobiles vont pouvoir interagir avec des acteurs fixes. Cet aspect est abordé plus en détail en 2.5.3.

Finalement, la mise en place de ces réseaux nécessite la connaissance de deux ensembles, qui peuvent différer en fonction de la vue du concepteur. D'une part, les paramètres sont des valeurs qui régissent le fonctionnement d'un réseau (tableau 1). D'autre part, les métriques sont des valeurs qu'il est possible d'obtenir sur le terrain, représentatives du comportement d'un réseau et de ses paramètres (tableau 2).

Echelle globale				
Util. Infra.	- Profil de l'utilisateur : politique de conduite, distribution des véhicules (type, taille), vitesse maximale.			
	- Contrôle central du réseau : surveillance et changement des paramètres globaux et locaux. - Coordination des équipements.			
Echelle locale				
Util.	- Gestion des priorités selon l'utilisateur : véhicules d'urgence, transport en commun, véhicule particulier (exemple : taxi). piéton.			
Infrastructure	- Voies, directions, priorités, chemins possibles. - Application de stratégies locales : cas des feux de circulation à une intersection où le réglage des feux verts et de leur durée est essentiel.			
	<table> <tr> <td>- Mise en place physique de détecteurs : type, nombre, position.</td> <td rowspan="2">} Échelle : réseau fixe.</td> </tr> <tr> <td>- Mise en place virtuelle de détecteurs : protocole de communication, métriques à prendre en compte, actions à effectuer.</td> </tr> </table>	- Mise en place physique de détecteurs : type, nombre, position.	} Échelle : réseau fixe.	- Mise en place virtuelle de détecteurs : protocole de communication, métriques à prendre en compte, actions à effectuer.
	- Mise en place physique de détecteurs : type, nombre, position.	} Échelle : réseau fixe.		
- Mise en place virtuelle de détecteurs : protocole de communication, métriques à prendre en compte, actions à effectuer.				
- Mise en place de capteurs mobiles (réseau véhiculaire) : <i>idem</i> .	} Échelle : réseau véhiculaire.			
	- Politique de gestion. - Politique de sécurité. - Échelle : système coopératif.			

TABLE 1 – Paramètres généraux des STI

Dans cette partie, nous avons défini des bases essentielles à la suite de ce document. Au cours de la prochaine partie, nous détaillons ce que sont concrètement les STI, leur étendue ainsi que le détail des technologies phares existantes.

Echelle globale									
Util.	<ul style="list-style-type: none"> <li>- Temps d'attente pendant le trajet et temps de parcours.</li> <li>- Profil de l'utilisateur :               <table style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding-right: 10px;">- Vitesse.</td> <td rowspan="2" style="font-size: 2em; vertical-align: middle;">}</td> <td>- Pollution.</td> </tr> <tr> <td>- Émission de carbone.</td> <td>- Sécurité.</td> </tr> </table> </li> </ul>	- Vitesse.	}	- Pollution.	- Émission de carbone.	- Sécurité.			
- Vitesse.	}	- Pollution.							
- Émission de carbone.		- Sécurité.							
Infrastructure	<ul style="list-style-type: none"> <li>- Temps minimum, maximum et moyen d'attente pendant le trajet (ATWT - Average Trip Waiting Time).</li> <li>- Vitesse maximale et moyenne des véhicules.</li> <li>- Pollution (pics, moyenne).</li> <li>- Fraudes et dangers (statistiques).</li> <li>- Congestion.</li> <li>- Trajets et itinéraires phares.</li> </ul>								
Echelle locale									
Utilisateur	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding-right: 20px;"> <ul style="list-style-type: none"> <li>- Détection de l'utilisateur à l'arrivée et au départ.</li> <li>- Nombre d'utilisateur par emplacement (exemple : voie).</li> <li>- OU Longueur d'une file à un emplacement et prise en compte éventuelle d'un espacement entre les utilisateurs.</li> </ul> </td> <td style="font-size: 3em; vertical-align: middle; padding-right: 10px;">}</td> <td style="padding-right: 20px;"> <ul style="list-style-type: none"> <li>- Débit.</li> <li>- Occupation.</li> </ul> </td> <td rowspan="2" style="font-size: 3em; vertical-align: middle; padding-left: 10px;">}</td> <td rowspan="2" style="vertical-align: middle;">Équité.</td> </tr> <tr> <td style="padding-top: 10px;"> <ul style="list-style-type: none"> <li>- Temps d'attente depuis le dernier feu vert.</li> </ul> </td> <td style="font-size: 3em; vertical-align: middle; padding-right: 10px;">}</td> <td style="padding-top: 10px;">Famine.</td> </tr> </table>	<ul style="list-style-type: none"> <li>- Détection de l'utilisateur à l'arrivée et au départ.</li> <li>- Nombre d'utilisateur par emplacement (exemple : voie).</li> <li>- OU Longueur d'une file à un emplacement et prise en compte éventuelle d'un espacement entre les utilisateurs.</li> </ul>	}	<ul style="list-style-type: none"> <li>- Débit.</li> <li>- Occupation.</li> </ul>	}	Équité.	<ul style="list-style-type: none"> <li>- Temps d'attente depuis le dernier feu vert.</li> </ul>	}	Famine.
<ul style="list-style-type: none"> <li>- Détection de l'utilisateur à l'arrivée et au départ.</li> <li>- Nombre d'utilisateur par emplacement (exemple : voie).</li> <li>- OU Longueur d'une file à un emplacement et prise en compte éventuelle d'un espacement entre les utilisateurs.</li> </ul>	}	<ul style="list-style-type: none"> <li>- Débit.</li> <li>- Occupation.</li> </ul>	}	Équité.					
<ul style="list-style-type: none"> <li>- Temps d'attente depuis le dernier feu vert.</li> </ul>	}	Famine.							
Infra.	<ul style="list-style-type: none"> <li>- Temps minimum, maximum et moyen d'attente (AWT / AJWT - Average Junction Waiting time).</li> <li>- Pollution locale".</li> <li>- Fraude et dangers (statistiques locales, prévention).</li> </ul>								

TABLE 2 – Métriques générales des STI

## 2 Les Systèmes de transport intelligents

### 2.1 Présentation

Le trafic routier urbain s'est amplifié en l'espace de quelques années, augmentant ainsi les problèmes engendrés qui sont nombreux et qui coûtent quotidiennement temps, argent, santé et qualité environnementale : embouteillages, accidents, pollution ou encore infractions. Pour l'exemple, une étude menée par IBM en juin 2011 ([55]) montre un passage de 8% (2010) à 28% (2011) de New-Yorkais ayant indiqué que les transports auraient grandement nuit à leur travail ou études. De même, à Moscou, par exemple, les conducteurs sont soumis à des embouteillages quotidiens de deux heures et demie en moyenne. Un handicap important qu'il devient de plus en plus nécessaire d'administrer.

La gestion du trafic routier s'inscrit dans le domaine des STI, qui visent à proposer des outils et modèles afin de gérer les aléas de ce dernier, ceci par le biais ou non d'équipements réactifs dits dynamiques. L'application de tels systèmes va avoir de multiples objectifs, parmi lesquels la fluidification du trafic, la détection d'incidents, la surveillance temps-réel du trafic, la diffusion d'informations ou de consignes variables aux automobilistes ou encore la réduction en conséquence de la pollution et des bruits.

### 2.2 Étendue

En ville, les STI s'étendent à de nombreuses applications. En premier lieu, ces derniers sont majoritairement conçus pour fluidifier et gérer le trafic routier, notamment au niveau des intersections où ces derniers peuvent directement agir sur les feux de circulation, également au niveau de la politique de stationnement, de l'information de l'utilisateur à tout niveau, et de l'utilisation de stratégies particulières afin de gérer les situations de danger. En second lieu, ces systèmes vont agir de manière plus ou moins directe sur des enjeux modernes tels que la pollution, en réduisant l'émission de gaz à effet de serre (conséquence d'une régulation cohérente du trafic). L'étendue des STI est immense, et se divise globalement en deux catégories dans la littérature : d'une part, ceux dont la contribution fait pleinement partie du domaine, et d'autre part, ceux qui y contribuent sans pour autant y faire référence : modèles purement théoriques, systèmes multi-agents, publications tantôt basées sur l'aspect matériel d'une technologie, tantôt sur les communications existantes.

### 2.3 Contexte et applications

**Les centres d'ingénierie et de gestion du trafic.** Les réseaux routiers urbains sont gérés par des *centres d'ingénierie et de gestion du trafic* (CIGT). Le rôle de tels organismes, généralement attachés à une zone géographique bien particulière, est de coordonner au mieux les différents éléments routiers, et de faire face aux situations de la vie de tous les jours (travaux, accidents, gestion du trafic routier et des pics, etc.). Ces centres, instaurés à la fin des années 90 en France ([1]), possèdent généralement un centre de contrôle permettant la manœuvre d'un ensemble de technologies placées sur le terrain. Ces systèmes sont au centre d'une infrastructure urbaine. Citons, par exemple, le *P.C Lutèce* situé au cœur de Paris, qui analyse environ 2000 équipements en quasi-temps réel ([2]). Les missions des CIGT sont multiples : recueillir les données en provenance directe des routes afin de remplir un rôle de superviseur et d'agir en cas de problème, gérer le trafic en cas d'imprévu, informer les usagers, assurer le suivi des événements et, d'une manière générale, s'assurer du maintien et du bon fonctionnement de l'ensemble des équipements du réseau. Remarquons que dans les pays anglo-saxons, la gestion du trafic est généralement divisée en plusieurs domaines et est gérée par des organismes tels que l'*Institute of Transportation Engineers* en charge des aspects suivants : caractéristiques du trafic, planification des transports, conception des infrastructures, contrôle, et maintenance organisationnelle / administrative / matérielle ([3, 51, 68]).

**Les intersections.** Le champ d'application des STI en milieu urbain est très large : en premier lieu, ces derniers agissent sur les intersections, en se chargeant d'appliquer une stratégie de changement des feux de circulation. Cette gestion des feux va représenter un aspect essentiel de la fluidité du trafic routier

dans une ville, et le problème est abordé par de nombreux auteurs, sous différents angles : **théoriques** (exemples : logique floue [61], réseaux de neurones [87, 80, 32] ou encore algorithmes génétiques [82, 40]), **pratiques** (exemple : mise en contexte avec placement de détecteurs [42]), **spécifiques** (exemple : étude de cas dans une résidence privée [63]), **techniques** (exemple : étude matérielle [69]), et d'autres aspects qui seront abordés plus en détail dans les parties suivantes.

La figure 1 montre le modèle de carrefour qui est typiquement utilisé dans la littérature afin de valider un modèle : une intersection composée de quatre directions avec un nombre fixé de voies pour chacune. Ici, les voies pour tourner à gauche sont séparées des voies allant tout droit ou à droite, ces deux derniers mouvements étant confondus. Ce modèle possède l'avantage de pouvoir être adaptable à de nombreuses situations, mais est instinctivement limité de part la distinction des mouvements et voies.

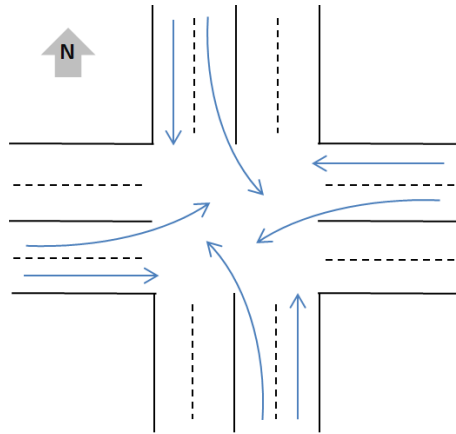


FIGURE 1 – Modèle de carrefour généralement utilisé dans la littérature

**Les voies spéciales.** Outre les intersections, les travaux insistent également sur la gestion des voies spéciales, où les transports en commun, taxi et pistes cyclables peuvent jouer un rôle important.

Dans [54] et [53], les auteurs discutent des possibilités existantes afin de détecter les bus, et de leur céder ou non la priorité en conséquence. Particulièrement, les possibilités suivantes sont listées : détection par l'infrastructure, par un centre de contrôle via signal GPS, par coopération avec une boîte de contrôle (échange d'un paquet identifiant le véhicule).

Les auteurs de [53] prennent en compte les arrêts que peuvent effectuer les bus : aux intersections mais également aux arrêts de bus. Les auteurs introduisent un système basé sur un détecteur de fermeture de porte allié à une détection par l'infrastructure, afin qu'un parcours donné soit correctement suivi.

Dans [57], les auteurs se penchent sur le cas d'un réseau de capteurs sans fil, et les utilisent au niveau des arrêts de bus afin de pouvoir calculer des éléments tels que le temps moyen de parcours ou encore la prédiction de l'arrivée d'un bus à un certain arrêt.

Dans [83], les auteurs proposent un modèle où chaque bus embarquerait un capteur : chaque ligne de bus est alors représentée par réseau véhiculaire, où des informations pourraient être échangées et des décisions prises (exemple : élection). Ce dernier modèle peut être généralisé à un réseau de taxis.

Enfin, [67] proposent d'utiliser un réseau de capteurs multi-sauts afin d'appréhender l'arrivée de bus au niveau d'un feu de circulation donné, et apporter de nouveaux paramètres à la décision de changement de feu à un carrefour. Notons que [37] reprend la même logique de fonctionnement, en généralisant les capteurs (qui ne sont pas assez répandus à l'époque de l'article) par des détecteurs.

Comme nous pouvons déjà le constater, la gestion des voies de bus est vue sous de multiples aspects, et la notion de détecteur est souvent symbolisée par des capteurs sans fil, souples et dynamiques vis-à-vis des tâches à gérer.

**Le stationnement.** La gestion du stationnement est également primordiale et a une incidence directe sur la fluidité du trafic : il paraît logique de dire que l'utilisation de la voiture en milieu urbain repose en partie sur le fait de savoir si oui ou non une place est disponible sur le lieu d'arrivée. Il faut savoir qu'en moyenne en France par exemple, 10% des véhicules en circulation à un instant donné cherchent une place où se garer ([24]). Les STI vont aider à prendre des décisions, mais également informer les utilisateurs ou encore contrôler les véhicules.

a) Exemple, des détecteurs peuvent être utilisés afin de détecter la présence d'un véhicule sur une place, et calculer sa durée de stationnement. Ceci a été constaté dans la ville d'Amiens via des stationnements "minute" : une borne est associée à une place, et dès lors qu'un véhicule s'y gare, un compte à rebours se déclenche pour une durée déterminée. Si cette durée est dépassée, les agents de la voie publique sont automatiquement prévenus.

b) Nous pouvons également citer l'utilisation de **panneaux à messages variables** (PMV) pour les parkings, systèmes très répandus dans les grandes métropoles qui indiquent le nombre de places disponibles (ceci n'utilisant pas nécessairement des détecteurs, mais étant généralement calculé en fonction des entrées / sorties dans le parking en lui-même).

**La sécurité.** Concernant la sécurité routière, les STI ont une grosse carte à jouer, principalement autour de deux catégories. D'une part, les systèmes **hors-véhicules** tels que les PMV vont permettre d'avertir l'utilisateur en cas de danger : vitesse d'un utilisateur trop élevée, conditions météo inadaptées, et travaux. Le but ici est d'influencer l'utilisateur. D'autre part, les **systèmes sur-véhicule** sont également nombreux et très développés : détection de piétons ou obstacles, capots intelligents capables de se soulever en cas d'impact avec un piéton ([4]), systèmes embarqués (exemple : appel des secours automatique en cas de choc), régulateur de vitesse, vision nocturne (à l'aide de caméras), etc.

**Les ronds-points.** Certains travaux abordent la gestion des ronds-points, en se basant sur les priorités existantes. Dans de nombreux pays, il est, par exemple, commun de donner la priorité aux usagers étant déjà à l'intérieur d'un rond-point.

Dans [66], les auteurs utilisent la mécanique des fluides (à base de théorie des flots et de méthodes de gestion du trafic) afin d'étudier ce que pourrait donner une gestion des priorités différente. Ceci, en partant d'un rond-point à trois directions avec une seule voie, pour arriver à un modèle généralisé à N directions de K voies. Les auteurs incorporent également des feux de circulation, afin de mesurer les avantages et inconvénients procurés par l'utilisation de ces derniers (par rapport à leur premier modèle). Ce qui en ressort apparaît comme une évidence : soit deux flots, chacun étant sur une voie respectivement au cœur du rond-point et sur les côtés, la priorité doit être donnée au flot de la plus grande importance afin d'obtenir une gestion optimale du trafic. Les feux quant à eux améliorent grandement la fluidité du trafic, mais uniquement en cas de débit suffisamment élevé.

Dans [77], les auteurs utilisent la théorie des files d'attente autour de quelques modèles afin de calculer le temps moyen d'attente des utilisateurs pour chaque intersection, avant d'entrer dans un rond-point : une file correspond ici à un ensemble de voitures sur une voie. Le calcul est effectué en fonction ou non de la présence d'un feu de circulation à chaque intersection : obtention d'une information à destination d'une éventuelle station de contrôle, de l'usager (PMV), pour le réglage des dits-feux.

Dans [97], les auteurs souhaitent gérer les flux de circulation d'un rond-point à l'aide de feux tricolores installés à des endroits stratégiques : les points où le croisement entre deux ensembles de véhicules est possible. Afin d'être efficaces, les auteurs proposent de baser le timing des feux en fonction d'une base de données historique, afin d'identifier les heures de pointes, et désactiver toute signalisation le reste du temps.

Enfin, citons [99], où les auteurs proposent d'analyser trois approches afin de fluidifier le trafic dans les ronds-points : avec des signaux de ralentissement à l'arrivée, avec des feux de circulation à l'arrivée, et avec des feux de circulation à la fois à l'arrivée mais également à l'intérieur du rond-point, lorsqu'un usager prend la voie de gauche. Il apparaît clair que la troisième méthode reste la plus efficace en cas de fort trafic, mais pas nécessairement dans les autres cas.

**Rapport à la pollution.** Concernant la pollution, les STI peuvent grandement aider à réduire l'énergie et l'émission des gaz à effet de serre provoquée par les véhicules, ceci au travers de plusieurs points.



Tout d'abord, la connaissance apportée par les STI peut influencer l'utilisateur à prendre les transports en commun, moins consommateurs et qui ont tendance à aller vers l'écologie (un bus émet 0.03 kg de CO<sub>2</sub> par passager-kilomètre, une voiture 0.11 ([72])).

Également, le développement des systèmes partagés va aider à prendre cette décision : vélo ou même voitures électriques (exemple : *Auto Bleue* à Nice ou *Mobility Car Sharing* en Suisse ([5, 6])). Dans ce premier cas de figure, il faut tout de même prendre garde au trafic induit : moins d'embouteillages et de contraintes peuvent décider un utilisateur à prendre sa voiture.

Ensuite, la gestion du trafic routier permet l'optimisation des déplacements et une viabilisation des temps de parcours : réduction des embouteillages en conséquence, des séquences d'arrêts et reprises qui sont les principales sources de consommation de carburant et d'émission de gaz à effet de serre.

Enfin, le comportement des conducteurs joue un rôle majeur sur la consommation de carburant. Au travers de systèmes informatifs (PMV, systèmes embarqués), les STI peuvent contribuer à l'apprentissage de valeurs écologiques auprès de l'utilisateur.

Pour finir, citons un projet prometteur : *Copenhagen Wheel* ([7]), qui vise à utiliser les vélos partagés dans la ville de Copenhague afin de récolter des données environnementales. Ceci est permis par l'utilisation de détecteurs incorporés au vélo, et fonctionnant de manière économique (utilisation de l'énergie du cycliste). L'objectif final de ce projet est de s'étendre à plusieurs villes afin de créer des bases de données environnementales, les informations seraient disponibles directement sur Internet ou Smartphone. Données de pollution, mais également diverses informations : manière de rouler, itinéraire ou même calories dépensées.

Ici, nous avons fait un rapide tour de ce que pouvaient proposer concrètement les STI en milieu urbain. Nous allons à présent nous concentrer sur un point primordial : la gestion des feux de circulation aux intersections, en abordant tout d'abord ce qui est fait classiquement sur le terrain, puis en faisant le tour des modèles proposés. La partie suivante est dédiée au cas particulier des réseaux de capteurs sans fil, déjà utilisés dans les quelques cas cités ci-dessus.

## 2.4 La gestion des feux de circulation

Dans cette partie nous orientons notre étude sur les systèmes de gestion des feux de circulation, en abordant des solutions et termes traditionnellement utilisés dans les zones urbaines.

### 2.4.1 Les contrôleurs de feux

De nos jours, les feux de circulation d'une intersection sont généralement gérés par une **boîte de contrôle**, qui va posséder plus ou moins de propriétés en fonction des constructeurs. Typiquement, une boîte est rattachée à une seule intersection et possèdent les éléments principaux suivants ([70]) :

- Une **unité d'énergie**.
- Une **unité de détection**, connectée à des éléments de contrôle (détecteurs).
- Une **unité de contrôle**, donnant l'ordre d'enclenchement des feux.
- Une **unité d'avertissement rapide**, réagissant en cas d'erreur critique (par exemple : orange clignotant sur l'ensemble des feux.).
- Une **unité de gestion des conflits**, qui est programmée avec les combinaisons de feux verts autorisés (**matrice de conflits**) et qui vérifie les données envoyées par l'unité de contrôle : elle fait appel à l'unité précédente en cas d'erreur ou de faute constatée sur l'un des feux.
- Une **unité d'administration**, pour prendre le contrôle du carrefour (par la police par exemple).

Afin de régir le fonctionnement des feux de circulation, les contrôleurs utilisent des mécanismes combinatoires et temporels bien spécifiques, dont les définitions sont ci-dessous.

**Définition 8.** Une **phase** représente un intervalle durant lequel une combinaison de feux verts autorisés par l'unité de gestion des conflits va être activée. Les phases sont déterminées à partir des mouvements que chaque direction peut effectuer.

**Définition 9.** Un cycle correspond à l'enchaînement d'un ensemble de phases. Ce cycle est typiquement fixé au minimum à 45 secondes et ne dure pas plus de 90 secondes ([86]) pour éviter de perdre du temps à arrêter et redémarrer le trafic. Classiquement, un cycle déroule l'ensemble des phases et mouvements possibles, de manière à ce que toutes les voies aient au moins une fois le feu au vert. Lorsque l'intersection est suffisamment équipée, cette règle n'est pas nécessairement applicable (par exemple, les voies sans véhicules peuvent ne pas être sélectionnées).

**Définition 10.** Un plan de feux correspond à la description d'un cycle donné, et définit les différentes phases que le cycle va dérouler ainsi que leur durée. Exemple donné à la figure 2 où le plan de feux est constitué d'un cycle de deux phases.

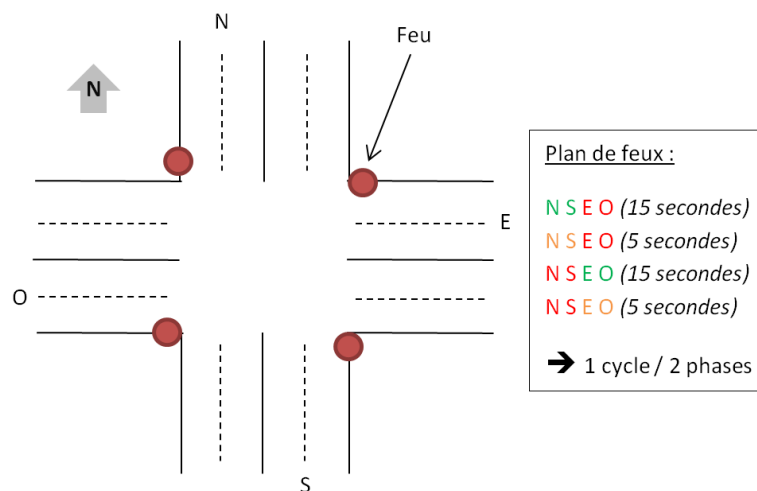


FIGURE 2 – Exemple d'un plan de feux

Les contrôleurs de feux peuvent être mis en place avec plusieurs modes de fonctionnement. Tout d'abord, les **contrôles prédéterminés**, où l'enchaînement des phases s'effectue en marge du trafic routier, c'est à dire toujours dans le même ordre et avec un temps prédéfini en fonction de l'importance connue des voies. Ensuite, les **contrôles semi-dynamiques**, où des détecteurs sont mis sur certaines voies (exemple : les voies jugées peu importantes) afin d'ajuster des paramètres tels que les temps de feux ou encore l'ordonnancement des phases. Enfin, les **contrôles dynamiques**, où des détecteurs sont mis sur l'ensemble des directions possibles. Il n'est pas forcément nécessaire de déployer des détecteurs à certains endroits : quelques auteurs ont donc essayé d'orienter les ingénieurs dans le choix d'un mode, par exemple [36] ou encore [79] décrivent leurs avantages et inconvénients, en fonction du matériel disponible et des données connues de l'intersection (exemple sur le tableau 3). Notons au passage l'existence du **contrôle par ordinateur**, où les phases sont ajustées à distance depuis des CIGT ou directement sur le contrôleur de feux (influence en cas d'imprévus, police, travaux, etc.).

Configuration du réseau	Charge de l'intersection ( $v/c$ )	Nombre de phases		
		2	4	8
Croisement	0.80	Prédéterminé	Semi-dynamique	Semi-dynamique
	> 0.80	Prédéterminé	Prédéterminé	Prédéterminé
Réseau dense	0.80	Dynamique	Semi-dynamique	Dynamique
	> 0.80	Prédéterminé	Semi-dynamique	Dynamique

TABLE 3 – Exemple d'orientation pour le choix d'un mode, en fonction de  $v/c$  (*vehicle-to-capacity*, la charge du réseau à l'endroit spécifié dans le tableau). Ceci est extrait de [79].

## 2.4.2 Nomenclatures

Face à la multitude de constructeurs existant dans la gestion du trafic routier urbain, et des spécifications souvent différentes amenant à des incompatibilités techniques et fonctionnelles, plusieurs organisations se sont regroupées dans le but de mettre en place des nomenclatures, afin de favoriser le développement des STI. Aux Etats-Unis par exemple, dès le début des années 1990, c'est la NEMA (*National Electrical Manufacturers Association*) qui s'est chargée de la rédaction de telles normes. Dans cette sous-partie, nous allons prendre exemple sur cet organisme, et détailler la logique de fonctionnement standard des contrôleurs de feux ([86, 48, 71]).

**Numérotation des phases.** L'une des normes rédigées par la NEMA définit la mise en place des phases possibles d'une intersection, en les numérotant en fonction de labels bien spécifiques affectés aux mouvements possibles, comme l'illustre la figure 3. Par exemple, les voies de la direction sud allant tout droit auront le label 6 d'affecté, celles de la direction ouest le label 8. La définition des phases va se faire en fonction de la numérotation des mouvements : plusieurs techniques existent, mais il suffit généralement de prendre le plus petit numéro affecté à un mouvement allant tout droit afin de nommer une phase. Exemple sur la figure 3 : si nous prenons la phase représentée par les mouvements bleus, soit les numéros 2, 5, 1 et 6, dont 2 et 6 correspondent à des mouvements allant tout droit, alors le numéro de la phase est 2. Pour une phase ne possédant pas de mouvement allant tout droit, alors le plus petit label parmi les autres mouvements est sélectionné. Notons qu'ici, les mouvements à droite ne sont pas représentés car confondus avec les mouvements allant tout droit (nous pouvons poser l'hypothèse que ces derniers reprennent les mêmes propriétés : si un mouvement de droite entre en conflit avec un autre mouvement, alors cela est nécessairement le cas pour le mouvement tout droit associé).

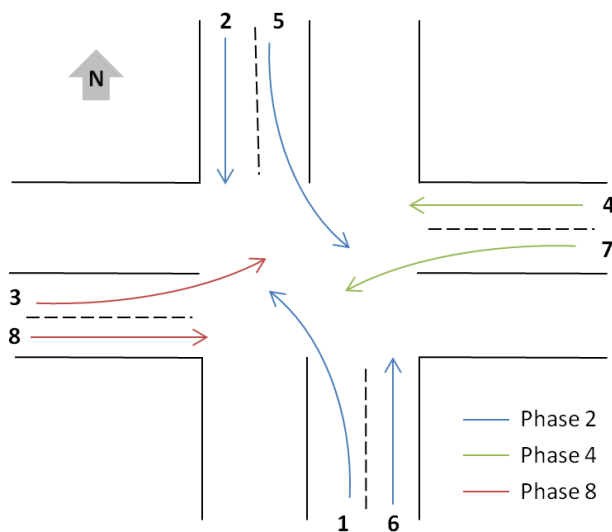


FIGURE 3 – NEMA - Affectation de labels aux voies, affectation des phases

Cette logique de numérotation n'est évidemment pas prise au hasard, et permet notamment de ne pas mettre certaines voies en conflit lors de la constitution des phases : certains mouvements ne pourront jamais être associés les uns avec les autres, comme par exemple le 8 et le 6, ceci pour des questions de sécurité minimale. Pour cette raison, des **diagrammes de phases** et **en anneaux** existent.

**Diagramme de phases.** Un **diagramme de phase** rassemble l'ensemble des phases d'un cycle, où chaque phase est représentée par un bloc de ce diagramme. Afin de dérouler un cycle sans conflit entre les phases, il suffit de dérouler ce diagramme. C'est notamment cette technique qui est utilisée dans le cas de contrôles à temps fixe.

Exemple : la figure 3 décrit une intersection possédant trois phases que nous nommons A, B et C : le diagramme de phases correspondant est représenté par la figure 4.

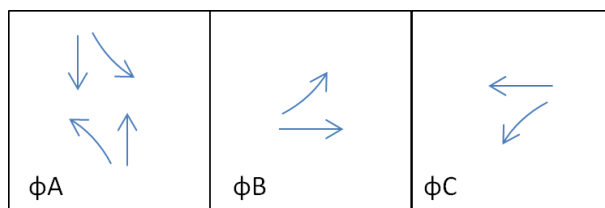


FIGURE 4 – NEMA - Diagramme de phases

**Diagramme en anneaux.** Un diagramme en anneaux permet de définir chacune des phases d'un diagramme de phases, en s'assurant qu'aucun conflit ne soit possible entre les mouvements existant. Les diagrammes en anneaux sont constitués de barrières qui isolent les mouvements appartenant aux rues antagonistes d'une intersection : les mouvements d'un côté de la barrière ne doivent pas se produire en même temps que les mouvements de l'autre côté de la barrière.

Exemple : en reprenant le schéma général d'une intersection à 4 directions, la barrière permet de séparer les 2 rues présentes.

Ainsi, si un diagramme en anneaux possède N barrières, il y aura N+1 phases possibles à constituer, une phase représentant un ensemble de blocs sans barrière. La sélection des mouvements de chaque phase va se faire à l'aide des anneaux présents de part et d'autre des blocs : chaque anneau va sélectionner un mouvement, l'assemblage de ces derniers forme une phase.

Exemple : sur la figure 5 représentant le cas d'une intersection classique, huit mouvements sont utilisés dans le diagramme en anneau, ce qui rend possible la création de plusieurs diagrammes de phases.

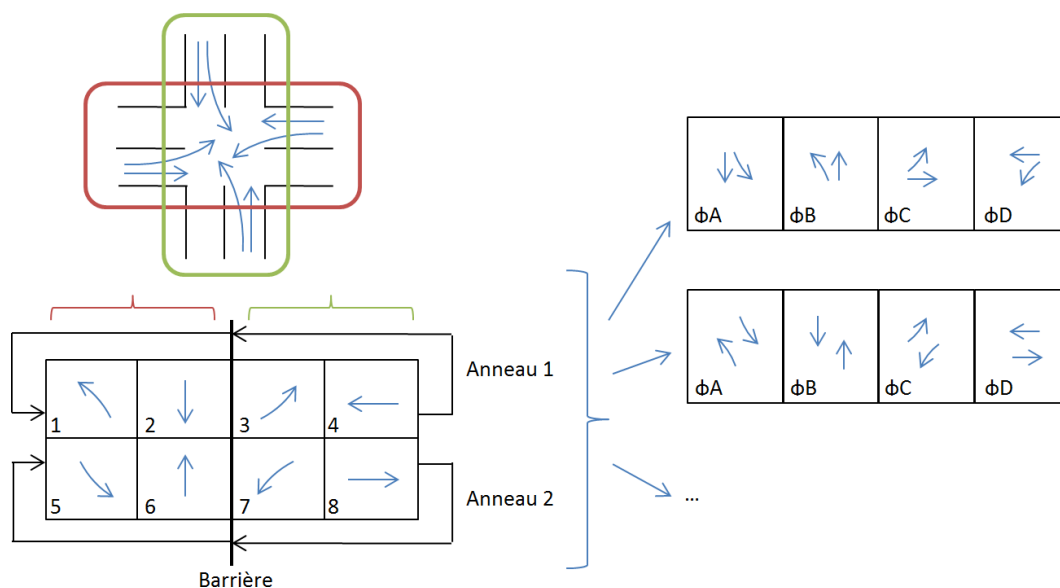


FIGURE 5 – NEMA - Diagramme en anneaux

**Nombre de phases.** Le nombre de phases dans un cycle est un élément important dans la fluidification du trafic routier : plus leur nombre est petit, plus la gestion du temps de feux et du cycle est souple et confortable, les arrêts de véhicules sont également plus rares. Cette minimisation est effectuée en fonction de la charge et de l'importance de certaines voies pouvant être en situation d'interblocage. Or, avec le modèle défini par les diagrammes en anneaux, aucune situation d'interblocage ne peut se

produire, le nombre de phases étant en conséquence au minimum de quatre. Toutefois, il est possible de changer légèrement ce mécanisme en utilisant uniquement la contrainte de la barrière comme séparateur de phase, ce qui nous limiterait non plus à quatre phases mais à deux si nous reprenons le cas d'un schéma général (figure 5).

Exemple : si les voies tournant à gauche sont peu chargées, il est possible de les confondre avec les autres mouvements. Le cas où ces voies seraient trop chargées peut amener à une situation d'interblocage, il nous faut donc respecter un schéma plus classique, et subdiviser l'ensemble afin d'arriver à quatre phases. L'établissement des phases est donc en premier lieu une question de charge sur les voies de gauche, et est réalisé en fonction de la souplesse souhaitée des concepteurs. Notons que cette charge peut être répartie différemment en fonction de son importance, comme l'explique [48].

Exemple : mouvements de gauche seuls, puis avec d'autres mouvements, puis uniquement les autres mouvements. A noter que selon [48], depuis plusieurs années déjà le nombre de phases est limité à huit dans la plupart des contrôleurs de feux, pour un schéma général (ceci pour ne pas provoquer trop de répétitions, d'arrêts, de rallongement du temps de cycle en conséquence).

Notons que le mécanisme est généralisable sur des intersections possédant un nombre de mouvements et directions différents.

Pour terminer, soulignons que la NEMA agit principalement aux Etats-Unis, certains pays possédant une logique similaire, comme le cas de la France avec la norme DIASER (*Dialogue standard pour les équipements de régulation*).

## 2.5 Vers une approche dynamique

### 2.5.1 Présentation

Le développement du trafic routier urbain a amené à concevoir des systèmes intelligents. L'une des particularités de ces systèmes est leur besoin de dynamisme et de réactivité : afin de pouvoir agir sur une situation, ces derniers ont besoin de connaître des informations sur ce qu'il se déroule à des endroits bien précis, et prendre des décisions en conséquence. Dans cette partie, nous allons étudier les particularités de tels systèmes, en présentant tout d'abord les particularités logicielles et matérielles de ceux se basant uniquement sur les données issues de l'infrastructure (détecteurs), puis en présentant plus vaguement des systèmes dits coopératifs, où des communications inter-véhiculaires vont être introduites. Nous finirons en décrivant des modèles théoriques souvent difficiles à appliquer en situation réelle. La gestion du trafic en milieu urbain opérant essentiellement au niveau des carrefours, c'est cet axe que nous allons continuer de développer.

### 2.5.2 Contrôle dynamique par l'infrastructure

#### Les systèmes de gestion classiques des feux de circulation

La gestion des feux de circulation est un problème qui a commencé à être étudié au début des années 1970 ([25]) avec l'apparition progressive de systèmes de gestion centralisés, en charge des carrefours d'une zone géographique donnée. Au fil des années, ces systèmes se sont diversifiés, et ont connu trois générations de contrôle. Aujourd'hui, ces générations peuvent être utilisées chacune en fonction des moyens mis en place sur l'infrastructure, et de la connaissance de cette dernière :

- **Première génération - contrôle à temps fixe** : en fonction de l'heure et parfois du jour, le système va utiliser un plan de feux prédéfini. Exemple : une configuration stricte est appliquée de 12h30 à 14h, tandis qu'une plus souple et équitable pour l'ensemble des voies d'un carrefour est appliquée à 3h du matin. Classiquement, trois configurations existent : matin, après-midi, et "reste" ([22]). Nous pouvons imaginer quelque chose de plus complet prenant en compte des plans spécifiques aux heures de pointe.
- **Deuxième génération - contrôle à temps dynamique** : des détecteurs sont utilisés afin de recueillir les données du trafic toutes les X minutes. Ces données peuvent être utilisées afin d'optimiser ou mettre en place un plan de feu. Par exemple, une fois les phases déterminées, l'un des enjeux va être de définir un temps de feu vert pour chacune d'entre elles : ce dernier est généralement

constitué d'une valeur minimale et maximale, afin de ne pas provoquer des arrêts intempestifs ou d'engendrer un cycle trop long. Le feu vert minimum suffit uniquement si pendant son temps d'exécution, aucun nouveau véhicule ne franchit un détecteur. Si un nouveau véhicule passe, un temps est ajouté pour ce véhicule, l'opération répétée éventuellement jusqu'au feu vert maximum.

- **Troisième génération - contrôle à temps réel** : reprend le même principe que la deuxième génération mais cette fois-ci en temps réel.

Notons que les deux dernières générations, qui introduisent un caractère dynamique au système, peuvent être chacune décomposées en deux types :

- **Contrôle réactif** : en fonction des données recueillies sur le terrain périodiquement (plusieurs minutes ou cycles), le système met en place une nouvelle configuration en réponse aux informations reçues. Cette méthode est le premier niveau de dynamique, et est simple à mettre en place, mais nécessite toutefois une très bonne connaissance du système afin d'être efficace. C'est également la première méthode à être apparue aux Etats-Unis à la fin des années 1980 avec l'apparition des UTCS (*Urban traffic control software*).
- **Contrôle adaptatif** : ce type de contrôle va programmer dynamiquement les plans de feux en se servant des paramètres recueillis sur le terrain, ceci en calculant des valeurs telles que le temps de cycle, des phases ou encore leur ordonnancement. L'opération va être effectuée de manière adaptative, c'est à dire en quasi-temps-réel. L'avantage de cette méthode est qu'elle peut s'adapter à de multiples situations, mais reste la plus compliquée à mettre en place (nécessité de cerner les informations à utiliser, comment les utiliser et se baser sur des théories parfois lourdes).

Les avancées ont été permises grâce à l'introduction de plusieurs solutions novatrices en termes de gestion du trafic routier : au total, ce sont plus d'une vingtaine de projets qui sont nés durant ces trente dernières années ([8]). Ici, nous allons essentiellement parler de deux systèmes qui ont retenu toute notre attention. Ces deux derniers ne sont pas nécessairement les plus performants, mais représentent les deux principaux systèmes de gestion du trafic routier utilisés dans le monde ([101]) :

**SCOOT.** Citons d'abord SCOOT ([74], *Split Cycle Offset Optimization Technique*), un système de contrôle à la fois réactif et adaptatif et entièrement centralisé, développé par le TRL (*Traffic research laboratory*, un centre de recherche anglais sur les transports). Dans ce système, l'ensemble des informations recueillies sur le terrain vont à un centre de gestion, qui s'occupe de traiter les informations et renvoyer des indications directement aux intersections. Les véhicules sont détectés par des dispositifs pouvant être placés à divers endroits sur les voies : au niveau des feux ou à une certaine distance afin de pouvoir mesurer le débit du trafic. SCOOT mesure en permanence le volume de véhicules de chaque côté de l'intersection et change la durée des phases en fonction d'un indice de performance, calculé par rapport au délai d'attente moyen, de la longueur des queues et des arrêts sur le réseau. Ainsi, de la même manière que TRANSYT ([38]), système sur lequel il est basé, SCOOT génère des plans de feux en fonction de la demande des utilisateurs (côté adaptatif). En plus de cela, le système utilise des informations en ligne issues de bases de données (historique ou autres, côté réactif). Notons au passage que SCOOT est l'un des principaux systèmes de contrôle adaptatif déployé mondialement ([9]).

**SCATS.** Citons ensuite SCATS ([78], *Sydney Coordinated Adaptive Traffic System*) qui a été à l'origine développé pour Sydney et d'autres villes Australiennes. Il est pour sa part entièrement adaptatif et utilise une notion de hiérarchie (ce qui forme une certaine distribution sur le réseau) : entre le recueil des données sur le terrain et le centre de contrôle, des contrôleurs intermédiaires sont insérés, permettant d'alléger la charge globale du système et d'avoir un contrôle découpé en plusieurs zones, l'ensemble des acteurs utilisant des communications synchronisées. De manière similaire à SCOOT, ce système ajuste le temps des cycles et autres paramètres en fonction des données recueillies afin de diminuer le délai et les arrêts, mais n'utilise pas la même stratégie : les valeurs recueillies permettent la sélection de plans de feux parmi une large librairie, sur lesquels le système va se baser pour proposer des plans adaptés. De plus, contrairement à SCOOT, les détecteurs sont uniquement placés au niveau des feux de circulation. Notons que les conventions utilisées par SCATS sont éloignées des standards NEMA, ce qui ne facilite pas son intégration sur les réseaux urbains ([25]).

Selon [76], l'installation de l'une de ces solutions prendrait en moyenne 365 heures (630h pour SCOOT), coûterait au total en moyenne 55.000\$ par intersection, et nécessiterait un temps d'entraînement moyen de 41 heures (dont 60h pour SCATS). Des chiffres considérables dont le prix peut être expliqué en premier lieu par l'équipement utilisé ([69]). Les auteurs de [44] estiment que par rapport à un système fixe, SCATS réduit globalement les temps de déplacement de 8%, les retards de 28% et les arrêts de 42%. SCOOT pour sa part réduit globalement les temps de déplacement de 8%, les retards de 22% et les arrêts de 17%.

**D'autres systèmes.** Tandis que SCATS et SCOOT sont conçus afin de ne prendre qu'une décision par cycle, d'autres systèmes plus modernes tels que OPAC ([47]), RHODES ([49]) ou encore InSync ([35]) - *Pour ne citer que eux* - analysent le trafic en temps réel et prennent des décisions seconde après seconde afin d'adapter dynamiquement un cycle. Nous pouvons également citer CRONOS ([33]) et Prodyn ([50]), les deux principaux systèmes agissant sur les carrefours Français ([45]). Le premier a été mis au point au début des années 1990 par l'INRETS et permet de dynamiser un carrefour en se basant notamment sur des images en temps réel en provenance de caméras, qui fournissent des informations telles que l'occupation de la chaussée. Le deuxième a été développé par le CERT et a la particularité de se baser sur des mécanismes classiques de cycles et phases.

De nombreux autres systèmes de ce type existent, nous n'avons cité que les principaux. Ces solutions "classiques" régissent à elles seules une bonne partie de ce qu'il peut exister en termes de gestion du trafic routier de manière complète et dynamique. Hélas, leur coût et leur durée d'installation constituent un frein à leur développement. Plus loin dans ce document, nous verrons de nouvelles approches de la littérature alliés à des principes davantage distribués et locaux qu'ici, pouvant s'allier à ces solutions et réduire en conséquence ces coûts globaux.

## Étude matérielle

Afin d'aller vers une approche dynamique, notamment au niveau des intersections, les logiciels et centres de contrôle ont besoin d'informations en provenance du terrain et recueillies à l'aide de détecteurs. Dans cette section, nous allons décrire brièvement les deux familles existantes. Les informations ci-dessous sont tirées essentiellement de [69].

**Les systèmes intrusifs.** Ces derniers nécessitent une installation souvent handicapante pour les usagers car situés généralement sur la chaussée. Les exemples les plus utilisés en pratique sont les boucles électromagnétiques, constituées de plusieurs spires de cuivre directement enfouies dans la chaussée : ces dernières permettent de mesurer les variations d'inductance lorsqu'un véhicule passe dessus. En clair, ce genre de dispositif va permettre de détecter des véhicules, de les classer en fonction de l'importance des variations, et même effectuer des mesures de vitesse (sous réserve d'avoir deux boucles séparées d'une distance connue). Après plusieurs comparatifs réalisés par [69], il s'avère que l'installation de ces boucles coûterait en moyenne de 500 à 800\$ à l'unité. Outre ces systèmes, nous pouvons citer les tubes magnétiques ou encore les réseaux de capteurs électromagnétiques sans fil, qui font l'objet de ce document et que nous détaillons dans la prochaine partie.

**Les systèmes non-intrusifs.** Ces derniers sont souvent volumineux et leur efficacité va dépendre des conditions environnementales, avec une installation généralement située sur le côté de la chaussée. Les matériels les plus fréquents sont les radars, posés en bord de route et pouvant être de plusieurs types : lasers, à ondes ultrasoniques, micro-ondes ou encore acoustiques. Des solutions tout aussi courantes sont les dispositifs de traitement d'images, filmant directement la circulation. Le traitement peut ainsi être automatique (exemple avec les logiciels open-sources [20] et [19] du même auteur) ou manuel (exemple : supervision humaine dans les centres de gestion). Selon [69], le prix de départ d'équipements fonctionnant par traitement d'images serait fixé à 5.000\$, et pourrait s'envoler jusqu'à 26.000\$ l'unité.

En dépit de la diversité apparente des solutions régissant le trafic routier depuis plusieurs années, la liste des inconvénients reste longue : prix de mise en place et de maintenance, durée d'installation, nécessité de câblage, solutions centralisées dans un centre de gestion. Les réseaux de capteurs sans fil, que nous aborderont à la partie 3., constituent une excellente solution à ces problèmes couramment rencontrés.

### 2.5.3 Systèmes coopératifs

Contrairement aux modèles reposant uniquement sur des capteurs électromagnétiques fixes, les systèmes coopératifs vont permettre une interaction entre plusieurs véhicules, ou entre plusieurs véhicules et l'infrastructure déjà en place. Ainsi, l'intelligence est répartie entre différents éléments du réseau, fixes ou mobiles, ce qui va nous permettre de développer des applications plus souples, plus efficaces et qu'il serait parfois impossible de mettre en place avec une architecture uniquement statique. Notons que le domaine des systèmes coopératifs est vaste et que nous n'en faisons ici qu'une introduction rapide.

**Les niveaux de coopération** Les avantages d'applications utilisant des capteurs mobiles et embarqués (voir partie 1.) sont nombreux : les données tirées des véhicules et de l'infrastructure sont généralement complémentaires, et trivialement, la réactivité d'un tel système est bien plus importante qu'un modèle démuné d'acteurs mobiles (le dynamisme est ici poussé à son extrême). Remarquons que ces systèmes peuvent être de plusieurs types en fonction du niveau de coopération que l'infrastructure est capable de posséder et de la volonté applicative encadrant le réseau ([21]), comme l'illustre la figure 6 :

- **Les systèmes autonomes (SA)**, où aucune communication n'est effectuée, mais où les véhicules disposent de capteurs embarqués leur permettant de mesurer des phénomènes présents autour du véhicule : détection d'accidents, d'obstacles, vérification de la distance de sécurité, etc.
- **Les systèmes coopératifs orientés infrastructure**, où un véhicule va pouvoir communiquer uniquement avec l'infrastructure (détecteurs, boîtes de contrôle), afin d'obtenir des données directement liées au trafic routier (I2V), ou afin d'en fournir (V2I).
- **Les systèmes coopératifs orientés véhicules**, où un véhicule va pouvoir communiquer uniquement avec d'autres véhicules, afin d'obtenir des informations spécifiques au trafic routier. Ici, nous pouvons imaginer deux types de communication :
  - **De proche en proche (V2V)**, où la communication est assurée uniquement entre deux véhicules voisins, à l'aide de capteurs.
  - **Communication globale**, où les différents acteurs possèdent un système relié à un oracle commun (exemple : Smartphones, certains GPS et autres).
- **Les systèmes interactifs (SI)**, où un véhicule donné va à la fois échanger des informations avec les autres véhicules, mais également avec l'infrastructure. L'avantage est considérable : outre la possibilité de traiter des informations agrégées de part et d'autres du système, les données importantes peuvent circuler au travers du réseau par l'intermédiaire des véhicules. La notion de partage prend ici tout son sens.

**Quelques travaux** Parmi les travaux existants, nous pouvons citer [75] qui décrit un réseau véhiculaire (V2V) : utilisation d'un réseau de capteurs sans fil allié à des communications Bluetooth afin d'améliorer la sécurité globale du réseau. Dans [89], les auteurs proposent un réseau de capteurs sans fil formé de trois types de nœuds : véhicules, capteurs électromagnétiques, et contrôleurs à une intersection. Les capteurs de route diffusent en permanence des informations contenant notamment leur position : les véhicules recevant des données de plus de trois capteurs différents vont alors pouvoir calculer leur position par triangulation et envoyer le résultat ainsi que leur vitesse au contrôleur, qui sera apte à prendre des décisions quant au changement des feux de circulation sur l'intersection (V2I). Dans [90], les auteurs reprennent un modèle similaire mais le développent à plusieurs intersections. A partir de ces deux travaux, les auteurs vont même jusqu'à proposer un prototype dans [39].

Certains auteurs ont étudié la possibilité d'un système multi-agents, un cas particulier de systèmes interactifs où les communications seraient faites entre les véhicules (les agents) et avec les feux de circulation. Parmi les travaux, citons [91], où Wiering propose des méthodes d'apprentissage permettant



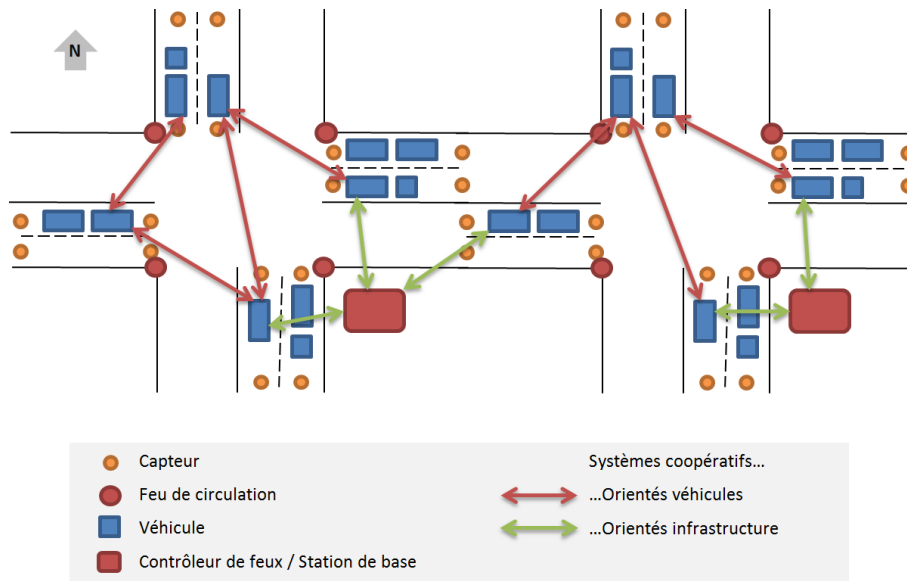


FIGURE 6 – Les systèmes coopératifs

aux véhicules de se déplacer en minimisant le temps d'attente aux intersections, en échangeant des informations avec les feux de circulation. Dans [92], l'auteur introduit la notion de vote : chaque véhicule estime ce qu'il a à gagner (son gain) si le feu auquel il se trouve passe au vert, et un vote est effectué parmi l'ensemble des agents présents à l'intersection. Dans [52], Houli et al. reprennent un modèle similaire à celui de Wiering qu'ils trouvent trop restrictif, et l'étendent à un modèle multi-objectif, afin de mieux gérer l'incertitude temps-réel du trafic : gestion des véhicules prioritaires, longueur des queues, etc.

En conclusion, il apparaît clair que les possibilités sont décuplées par rapport à des systèmes plus fixes et traditionnels. La mise en place reste toutefois problématique : nous ne pouvons pas forcer l'ensemble des usagers à utiliser une technologie. Ceci nous impose donc soit de nous baser sur un modèle utopique, soit sur un modèle sélectif, où seuls les usagers bénéficiant de la dite technologie pourront intégrer le système.

#### 2.5.4 Outils théoriques

Il est courant pour des modèles dynamiques de la littérature de se servir d'outils théoriques, parfois en faisant un rapprochement à la réalité, parfois sans aucune notion physique (technologie utilisée, disposition). Ici sont étudiés plusieurs outils, répandus dans la littérature et servant de base à certains modèles étudiés plus loin dans le document.

### Intelligence artificielle

#### Contrôle par logique floue

La logique floue permet de mettre en place des degrés dans la vérification d'une condition, et non plus se borner à un choix strictement binaire. Ce principe est utilisé par quelques auteurs pour traiter le problème de la gestion des feux de circulation et permet de simplifier le problème, ce qui change des méthodes d'optimisation mathématique habituelles souvent lourdes. Quelques exemples de travaux peuvent être trouvés au travers de [61, 26]. Nous pouvons également citer [102] qui utilise la logique floue afin de déterminer le temps d'un feu en fonction du nombre de véhicules présents sur les voies : à un nombre de véhicules correspond un intervalle définissant une durée de feu (exemple : moins de 5 véhicules par minute donne le feu vert pendant 10 secondes). Ce principe apparaît comme idéal à utiliser :

- Théorie simple s’appliquant à des problèmes complexes.
- Aucun modèle mathématique requis.
- Robustesse de la commande floue par rapport aux incertitudes.

Les inconvénients sont tout de même importants ([62]) : les techniques de mise en place et les réglages sont empiriques et aucune théorie ne permet de démontrer la stabilité et la robustesse d’une telle méthode.

### Algorithmes génétiques

Les auteurs de [40] ou encore [82] ont proposé d’optimiser le temps à une ou plusieurs intersections en se basant sur un algorithme génétique. Le principe est le suivant : obtenir une solution approchée d’un problème d’optimisation lorsqu’il n’existe pas de méthode exacte afin de le résoudre en un temps raisonnable. Dans ce type d’algorithme, la solution est approchée par bonds successifs (mutations). Ainsi, en ayant connaissance du nombre de véhicules et du temps moyen d’attente à une intersection, l’algorithme va pouvoir appliquer des méthodes d’optimisation et les améliorer au fil de sa vie. Si l’idée paraît intéressante, les contraintes sont évidentes pour le cas du trafic routier :

- De nombreux calculs sont nécessaires.
- Paramètres difficiles à déterminer, et il peut y avoir un certain délai avant l’obtention d’un résultat réellement efficace.
- Il est impossible d’assurer qu’une solution trouvée soit la meilleure, même après une multitude de mutations.

### Réseaux de neurones

Les réseaux de neurones sont inspirés du fonctionnement des neurones biologiques, et mettent en œuvre l’apprentissage par l’expérience. Dans le cas du trafic routier, plusieurs auteurs se sont penchés sur ce schéma (exemples : [87, 80, 32]). Ici, il est question d’effectuer rapidement des classifications, et d’apprendre à les améliorer, plutôt que de passer par un schéma traditionnel de modélisation. La logique floue et les algorithmes génétiques peuvent être vus comme des compléments aux réseaux de neurones ([87]).

### Théorie des files d’attente

La théorie des files d’attente est particulièrement adaptée au cas de la gestion du trafic routier : cette dernière appartient au domaine des probabilités, et permet une gestion optimale des files d’attente (ou queues). Dans le cas des intersections, une file d’attente est automatiquement créée lorsque les véhicules (clients) souhaitent obtenir un feu au vert (serveur). Particulièrement, il est facile avec cette théorie de calculer des valeurs telles que le nombre moyen de véhicules en attente, en service, le temps moyen d’attente ou encore de séjour dans le système.

Afin de décrire un système utilisant une file d’attente, la notation de Kendall est généralement utilisée. Cette dernière peut se résumer par une suite de trois symboles  $a/s/C$ . Où  $a$  représente la loi de probabilité des instants d’arrivées et  $s$  de la durée de service (au feu), qui est généralement exponentielle ( $M$ ) ou générale ( $G$ ).  $C$  représente pour sa part le nombre de serveur (1 dans le cas d’une voie).

Si nous reprenons les métriques présentées dans la première partie, certaines valeurs vont être faciles à calculer à l’aide de cette théorie :

- Soit  $\lambda$  la fréquence moyenne d’arrivées et  $\mu$  la fréquence moyenne de service.

- Soit  $C = \frac{\lambda}{\mu}$  la charge du système.

- Prenons le cas d’une file  $M/M/1$ . Le temps moyen d’attente serait représenté par  $\frac{C}{\mu \cdot (1 - C)}$ . Le

nombre moyen de véhicules en attente par  $\frac{C^2}{1 - C}$ . Le temps moyen de séjour dans le système par

$$\frac{1}{\mu} \cdot \frac{1}{1 - C}.$$

L'inconvénient de cette théorie est qu'elle nécessite un certain nombre de points de mesure dans le cas de la gestion d'une intersection. Dans la partie 2, nous verrons que les réseaux de capteurs sans fil peuvent facilement s'adapter son utilisation, et étudierons des modèles tels que [96], utilisant pleinement les principes décrits ci-dessus.

### Mécanique des fluides

La mécanique des fluides est un outil plus rarement utilisé dans le cadre du trafic routier, mais est tout de même à citer : ce dernier représente l'étude du comportement des fluides, dans notre cas des fluides en mouvement (on parle alors de dynamique des fluides). Cet outil n'est que peu utilisé dans le cadre des intersections, mais revient souvent lorsqu'il s'agit d'étudier des systèmes sans feux de circulation, c'est à dire se basant uniquement sur des priorités. L'exemple que nous pouvons citer est le cas des ronds-point (qui a été abordé précédemment en 2.3), où [66] décrit les flux de circulation comme étant des fluides en opposition. Le constat établi est que généralement, le plus gros fluide prend le dessus (et donc la priorité) sur le flux de plus faible importance.

Ce procédé est original à utiliser, et agréable dans certains cas. Hélas, il ne suffit pas à décrire le comportement individuel de chaque véhicule, et apparait comme étant inadapté à des cas où des feux de circulation sont présents.

## 2.6 Vers une approche décentralisée

Comme étudié, les réseaux routiers urbains sont gérés par des centres de contrôle. Le rôle de tels organismes, généralement attachés à une zone géographique bien particulière, est de coordonner au mieux les différents éléments routiers, et de faire face aux situations de la vie de tous les jours (travaux, accidents, gestion du trafic routier et des pics, etc.). Ces centres permettent la manœuvre d'un ensemble de technologies placées sur le terrain. En particulier les dispositifs de supervision et autres outils type SCAT ou SCOOT. Face à des systèmes centralisés de ce type, l'adoption progressive de systèmes distribués va représenter un énorme gain. D'une manière générale, nous pouvons dire que les systèmes distribués représentent les avantages suivants :

- **Coût** : plusieurs processeurs (détecteurs) à petits prix au lieu d'une grosse unité de calcul, si ce n'est d'avantage.
- **Performances de calcul** : les moyens centralisés sont généralement moins efficaces qu'une approche décentralisée, ou alors à très fort coût. De plus, le calcul parallèle devient naturel.
- **Sécurité** : une panne ou faute matérielle ou logicielle sur un système centralisé peut être fatale à ce système.
- **Évolution** : facilitée par l'auto-organisation naturelle d'un système distribué.

Dans le cas plus spécifique des STI, nous pouvons citer [85] qui indique que les approches centralisées ne peuvent pas faire face à la complexité croissante des réseaux de circulation urbains. En particulier, l'utilisation d'un réseau décentralisé permet d'améliorer la gestion de la circulation localisé sur une zone bien précise (exemple : intersection), et permet également gestion plus souple de la communication entre éléments voisins.

Remarquons que dans l'immédiat, un centre de contrôle global existera toujours, la machine n'étant pas capable de se passer totalement de l'homme. L'application de technologies distribuées peut néanmoins amener ce centre de contrôle à recueillir moins de câblage, moins de ressources en termes de calcul, moins de moyens.

Dans cette première partie, nous avons étudié ce qui se faisait traditionnellement dans les STI, et avons cerné plusieurs points intéressants : la nécessité de réactivité et de dynamisme, de distribution, ou encore l'intérêt croissant des équipements à faible coût. Dans la partie suivante, notre étude va se

recentrer sur notre intérêt principal : l'application de réseaux fixes de capteurs sans fil aux STI, qui apparaissent comme étant une solution idéale à ce genre d'applications. Nous verrons alors quelques approches se reposant sur une technologie distribuée et en plein essor.

## 3 Gestion du trafic routier par un réseau fixe de capteurs sans fil

### 3.1 Présentation

Comme nous l'avons vu, les réseaux fixes de capteurs sans fil sont un cas particulier des réseaux ad-hoc : ces derniers possèdent typiquement des contraintes supplémentaires, les rendant massivement accessibles et à faible coût. De tels capteurs représentent une solution idéale dans le cas de la gestion du trafic routier, ceci en comparaison avec les équipements traditionnels : [43] montre par exemple que ces dispositifs sont plus efficaces que les boucles électromagnétiques, de part leur réactivité, facilité de mise en place et nombre de points de mesure. Particulièrement, le coût et le temps d'installation de tels réseaux sont très confortables, comme le montre [69] ou encore [58], qui décrit la fabrication d'un capteur pour la somme de 27.3\$.

Précisons également que les capteurs ont l'avantage d'être petits (comparable à une pièce de monnaie selon certains ([58]) et peuvent donc s'adapter à de nombreux environnements, à des endroits où les dispositifs classiques ne pourraient parfois pas être installés. De plus, la coordination est possible avec une ou plusieurs stations de base généralement en charge des communications extérieures ou de l'administration du réseau. Dans notre cas, on pense immédiatement à une interface avec les contrôleurs de feux. Enfin, insistons sur les contraintes que représentent les capteurs (énergétiques, sécurité (sans-fil), mémoire et calculs limités, partage du spectre radio) : les modèles établis sur ce type de réseau doivent prendre en compte ces facteurs, tout en maximisant l'efficacité de la solution proposée.

Cette étude s'organise comme suit : tout d'abord, nous détaillons les principes de fonctionnement des capteurs électromagnétiques (la manière dont ces derniers peuvent détecter des véhicules), puis nous faisons un état de l'art de la littérature utilisant de tels réseaux pour la gestion des feux de circulation (pilier de guerre en zone urbaine), en catégorisant l'ensemble en fonction de l'infrastructure impliquée : simple intersection, multiples intersections, sur des applications particulières. Enfin, nous terminons l'étude par une comparaison des solutions proposées.

### 3.2 Principe de fonctionnement

Les capteurs électromagnétiques se comportent de manière similaire aux boucles du même nom : lorsqu'un large objet métallique (véhicule) passe par-dessus un capteur, ce dernier enregistre les variations produites sur le champ magnétique terrestre. Ces variations permettent de détecter un véhicule, de connaître son type (en fonction de l'intensité des variations), de mesurer sa vitesse ou encore sa longueur (estimées ou avec deux capteurs séparés d'une distance connue).

Plus précisément ([42, 41, 69]), les matériaux ferreux des véhicules vont produire un dipôle magnétique particulier, entraînant des perturbations sur le champ magnétique terrestre à priori stable qui sont mesurées par le magnétomètre du capteur : ceci a pour effet l'enregistrement éventuel d'un signal de détection. Ces perturbations sont mesurées sur trois axes mais seul l'axe de la hauteur est pris en compte (Z), l'axe des abscisses (X) et des ordonnées (Y) pouvant être troublés par la présence d'un second véhicule situé sur une autre voie ou dans la même file. L'ensemble de ces principes est illustré par la figure 7. Après une série de tests, les auteurs de [41] remarquent une fiabilité très grande quant à la comptabilisation du nombre de véhicules (99% constatés) et de la mesure de la vitesse moyenne (90% constatés).

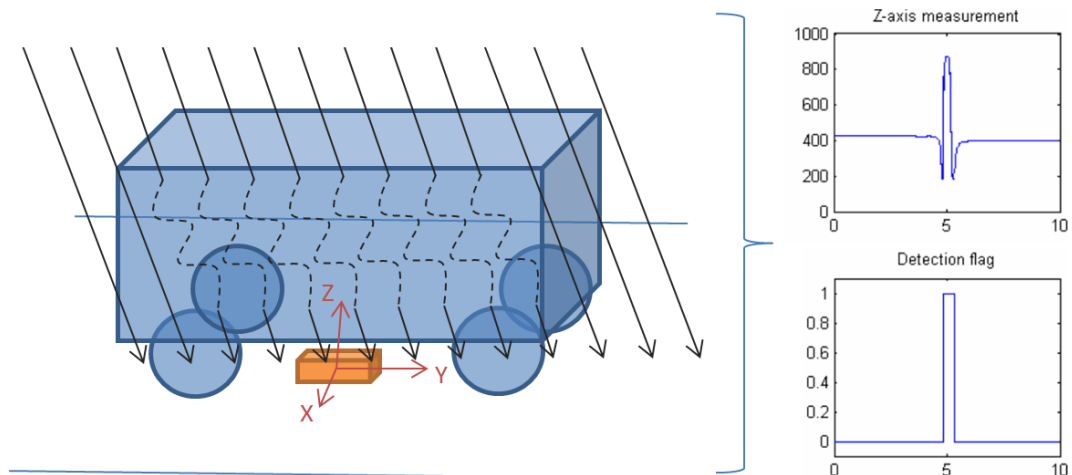


FIGURE 7 – Détection d’un véhicule par un capteur

Les mesures sont des exemples issus de [41] : le premier graphique affiche les variations brutes recueillies par un magnétomètre échantillonné à 128Hz au passage d’un véhicule, la deuxième figure correspond à l’enregistrement d’un signal de détection : ce dernier apparaît lorsque 10 échantillons successifs sont enregistrés et dépassent un certain seuil.

### 3.3 État de l’art

Dans cette partie, nous étudions la manière dont les différents auteurs utilisent les capteurs électromagnétiques en milieu urbain, notamment pour la gestion des intersections et des feux de circulation. Cette étude est décomposée suivant les différentes infrastructures sur lesquelles les modèles se basent : une intersection (en reprenant le schéma général de la figure 1), plusieurs intersections, et enfin sur des applications particulières.

#### 3.3.1 Sur une intersection

De nombreux auteurs se sont penchés sur la modélisation d’une intersection dotée de feux de circulation, base de la régulation du trafic routier urbain. En fonction des cas, des théories et pratiques plus ou moins ingénieuses ont été mises en place, nous étudions ici les principaux travaux.

#### Modèle général

La littérature se reposant sur les réseaux de capteurs sans fil pour gérer les feux de circulation se base généralement sur un modèle similaire, que nous décrivons ici. L’infrastructure utilisée est la même que celle représentée sur la figure 1 en faisant l’hypothèse que chaque voie possède un capteur après le feu (enregistrement des départs) et un autre avant le feu (enregistrement des arrivées, à une certaine distance). De plus, une station de base est présente sur le côté de la chaussée afin de recueillir les données des capteurs, celle-ci étant rattachée au contrôleur de feux traditionnellement présent. La figure 8 représente cette infrastructure.

Ce modèle sert de base à l’application d’algorithmes et de principes ayant pour but de fluidifier au maximum le réseau, en appliquant une gestion cohérente et efficace des feux de circulation. Ceci, de manière dépendante des véhicules étant détectés par les capteurs électromagnétiques. Ci-dessous, nous étudions les propositions de la littérature.

#### Algorithmes de gestion des feux

L’utilisation de la théorie des files d’attente est une approche venant en tête naturellement lorsqu’il s’agit de modéliser les queues présentes aux intersections. Les auteurs de [28] abordent par exemple son utilisation, qui permet d’obtenir à partir des données recueillies des capteurs la taille moyenne d’une queue

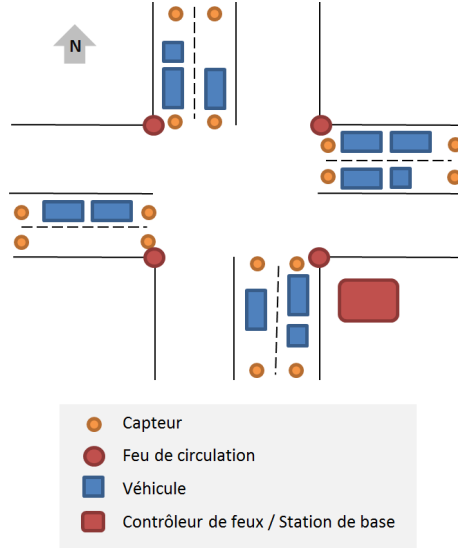


FIGURE 8 – Modèle sur une intersection généralisée avec 2 capteurs par voie

sur une voie et en conséquence le temps de feu vert noté  $T_G$ , décrit comme étant  $T_G = \min(T_s + \Delta, T_{max})$ , avec  $T_{max}$  le temps de feu maximum autorisé,  $T_{serv}$  le temps de service nécessaire à vider les queues présentes, et  $\Delta$  un temps variable. Parallèlement, d'autres auteurs ont proposé une approche plus aboutie dans [96] se reposant sur le modèle décrit ci-dessus (en fixant à huit voitures la distance des capteurs avant le feu), nous allons nous attarder sur cette proposition.

Sur l'intersection décrite par la figure 8, il existe exactement douze mouvements possibles : trois par direction (gauche, tout droit, à droite). Dans [96], les auteurs utilisent la théorie des files d'attente pour calculer les combinaisons de directions les plus intéressantes à traiter : chaque mouvement possible est modélisé comme étant une file M/M/1 avec des arrivées aléatoires et un temps de service exponentiel (le serveur représentant le feu). Concrètement, l'algorithme de Yousef et al. se décompose en deux parties. D'une part, un algorithme global nommé TSCA qui va se charger de l'initialisation des slots de temps et de méthodes garantissant la communication entre tous les éléments du système, la mise en attente du recueil des données et le lancement de l'algorithme d'optimisation de phases et de temps de feux. D'autre part, ce dernier algorithme qui utilise la loi de Little afin de déterminer les longueurs de queues  $Q_L$  ainsi que le temps moyen d'attente sur une file  $W = \frac{Q_L}{\lambda}$ , avec  $\lambda$  la fréquence moyenne d'arrivée des véhicules suivant une distribution de Poisson ([23]). Il est montré que la longueur des queues pour une file donnée varie en suivant l'équation :

$$Q_{L_j} = Q_{L_{j-1}} + \lambda_G G - \mu_G G + \lambda_R R$$

Avec  $j$  le numéro de cycle courant,  $G$  le temps de feu vert pour une phase,  $R$  le temps de feu rouge (en toute logique,  $G$  et  $R$  cumulés représentent un cycle de temps  $T$ ), et  $\mu$  la fréquence moyenne des départs.  $Q_{L_{j-1}}$  représente donc la longueur de queue au cycle précédent,  $\lambda_G G$  les véhicules arrivés pendant le feu vert,  $\mu_G G$  les véhicules partis pendant le feu vert, et enfin  $\lambda_R R$  les véhicules arrivés pendant le feu rouge.

A l'aide de ces informations, l'algorithme est en mesure d'élaborer un plan de feu dynamique cycle après cycle et composé de quatre phases au maximum, de manière à ce que la file d'attente moyenne et le temps d'attente moyen soient le plus petit possible ; ceci, en se basant sur l'ensemble des combinaisons possibles des douze mouvements existants, respectant une matrice des conflits permettant de ne pas créer d'interblocages. Ainsi, les phases  $i$  seront sélectionnées parmi les combinaisons offrant les plus grandes queues accumulées  $max_i$ , calculées par  $\sum_{i=1}^{12} Q_j^i$ , pour  $j$  allant de 1 à  $r$  (configurations possibles selon la matrice des conflits).

Finalement, le temps de feu vert pour la phase  $k$  - noté  $G_k$  - est déterminé de manière proportionnelle

aux longueurs de queues précédemment obtenues, à l'aide de la formule suivante (en supposant 4 phases) :

$$G_k = \frac{max_k}{max_1 + max_2 + max_3 + max_4} \cdot T$$

Cette solution offre de très bons résultats en utilisant une méthode à priori adaptée au cas des intersections. Toutefois, des problèmes existent. En premier lieu, la distance des capteurs situés avant les feux est fixe, ce qui peut poser problème lors de la mise en place. Ensuite, bien que l'algorithme balaye un large choix de possibilités en termes de phases, il n'est pas capable de traiter des cas tels que la phase 2 de la figure 3 (conflits minimales possibles). Enfin, le modèle repose principalement sur les plus grandes queues présentes, il pourrait être intéressant - au vu de la complexité du trafic routier - de prendre en compte d'autres paramètres, tel qu'un niveau de faim permettant de ne pas mettre de côté les voies de faible importance (voir métriques partie 1.).

Outre les files d'attente, d'autres articles utilisent des principes intéressants, toujours en se reposant sur l'infrastructure décrite à la figure 8. Après avoir abordé un modèle distribué où les capteurs communiquent avec une unité de contrôle d'intersection dans [85], Tubaishat et al. proposent dans [84] une gestion des feux plus aboutie. Les auteurs ont ainsi listé différentes combinaisons d'état des feux (rouge, flèche à droite, flèche à gauche, vert, etc) et ont établi trois plans de feux fixes, avec respectivement 4, 6 et 8 phases, sur lesquels ils effectuent des simulations. La solution consiste à se baser sur un système de gain, qui correspond pour une voie au nombre de voitures entre les deux capteurs. Ainsi, en fonction du plan de feu, chacune des phases va posséder un gain, somme des gains des différents mouvements : celle qui sera la plus chargée sera allégée en premier, jusqu'à ce que sa charge passe en dessous d'une autre ou que le temps de feu maximum soit dépassé. Le simulateur ici utilisé est GLD (voir section 3.), conçu afin d'optimiser les temps de parcours et particulièrement la gestion des contrôleurs de feux. Après quelques simulations, les auteurs arrivent à la conclusion intéressante qu'un modèle reposant sur deux capteurs par voie est plus efficace qu'un modèle se reposant sur un seul capteur. L'utilisation d'un système de gains afin de fluidifier le trafic routier est intéressant, cependant, de nombreux paramètres restent fixes, tels que les phases en elles-mêmes. De plus, l'utilisation de GLD rend irréaliste ce modèle : le changement d'une phase peut intervenir sans temps minimum (*voir plus bas*).

Zhou et al. ([101]) proposent un algorithme de contrôle des feux adaptatif en se basant sur le modèle établi à la figure 8, à l'exception que le placement des capteurs électromagnétiques est dynamique : ces derniers sont placés par rapport au temps de feu qu'il est possible d'avoir au maximum, ceci afin de détacher au possible l'efficacité de l'engorgement du trafic. La constitution du plan de feu se repose sur les combinaisons de mouvements qu'il est possible d'effectuer simultanément et sans conflits, c'est-à-dire douze comme l'illustre la figure 9. Cette solution va choisir la séquence de feux adaptée parmi ces combinaisons ainsi que la durée pour chacun des éléments de la séquence, en estimant le temps nécessaire à vider la plus grande file (ce temps étant borné par une valeur maximale). Dans ce modèle, tous les véhicules sont du même type de longueur  $L_{veh}$ , et chacun roule à la même vitesse  $V_{veh}$ . Le principe est d'utiliser un ensemble de facteurs afin de déterminer un plan de feux cycle après cycle, ces derniers étant fixés par priorités :

- Sont tout d'abord pris en compte les combinaisons possédant une ou plusieurs **voies avec des impératifs** (exemple : détection d'un véhicule d'urgence).
- Sont ensuite pris en compte les **blancs**, où un blanc de longueur  $L_{Blanc} = T_{Blanc} * V_{veh}$  est enregistré lorsqu'un capteur ne détecte aucun passage de véhicule pendant le temps  $T_{Blanc} > \frac{L_{veh}}{V_{veh}}$ . Plusieurs cas se distinguent alors. Si toutes les combinaisons possèdent au moins un blanc, on sélectionne celle pour qui le premier blanc enregistré est le plus court. Ainsi, si cette combinaison est sélectionnée, davantage de voitures seront susceptibles de passer. Si un cas ou plus n'a pas de blanc, alors on passe au critère suivant.
- Puis le **niveau de faim**, qui va permettre la sélection d'une combinaison de voie si cette dernière n'a pas été sélectionnée depuis un certain temps et qu'au moins une voiture y attend.
- Puis la combinaison ayant le **temps d'attente** le plus élevé.
- Enfin la combinaison ayant la plus grande file.

Afin de valider leur modèle, les auteurs utilisent une plateforme qui leur est propre, baptisée iSens-

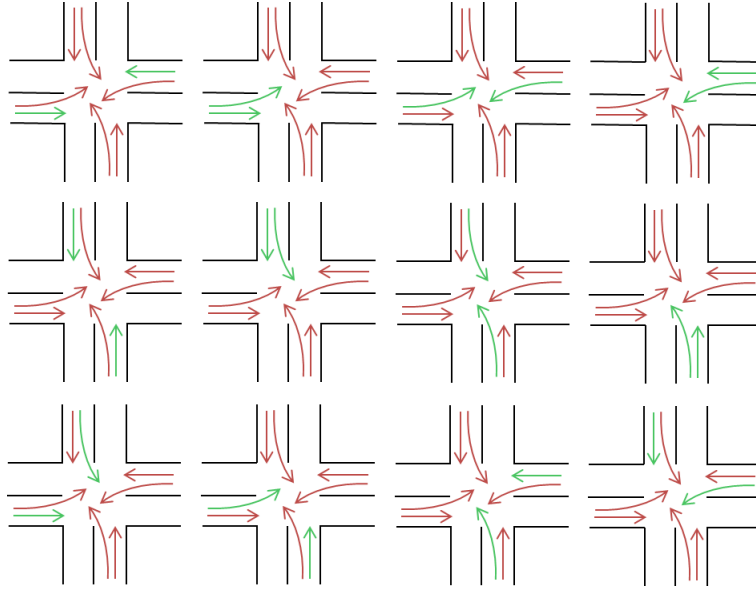


FIGURE 9 – Combinaisons sans conflits possibles pour un carrefour à 4 directions

Net ([34]). La solution proposée par Zhou et al. possède l’avantage de prendre en compte une distance dynamique pour le positionnement des deuxièmes capteurs, et ajuste à la fois les phases et le temps de ces dernières. Toutefois, tout comme [96], l’ajustement des phases reste borné à des mouvements sans conflits : ici ne sont par exemple pas pris en compte le cas d’une phase où certaines voies seraient peu encombrées, rendant possible une configuration avec conflit. De plus, la prise en compte du temps de feu maximum peut parfois amener à des problèmes, ne serait-ce que pour la distance irréaliste possible du deuxième capteur. Enfin, une grosse partie de la solution repose sur la détection de **blancs** : ceci n’est hélas permis que par l’utilisation d’hypothèses irréalistes, imposant l’homogénéité des véhicules et de leur vitesse.

Zou et al. ([102]) proposent une approche intéressante et afin de mesurer les arrivées : ces derniers utilisent un modèle différent en incorporant uniquement un capteur par direction (soit quatre pour une intersection) et posent l’hypothèse réaliste que ces derniers sont capables de détecter les variations du champ magnétique terrestre sur cinq mètres, une route étant constituée de deux voies, soit 6.5m de large. Ainsi, en mettant un capteur en bordure de route avant chaque feu de l’intersection, leur modèle est capable de comptabiliser le nombre de voitures empruntant chacune des voies. Cet unique paramètre permettant d’établir la durée des feux selon des intervalles : par exemple, si le nombre de passage est inférieur à cinq par minute sur une direction, alors le feu correspondant durera dix secondes. Ceci est une application de la logique floue. Les auteurs définissent également un mécanisme d’économie d’énergie propre aux capteurs : lorsqu’une direction possède son feu au vert, alors le capteur correspondant se met en veille, et se réveille lorsque le feu passe au rouge, afin de comptabiliser les arrivées. Enfin, les auteurs utilisent des données directement extraites sur le terrain afin de simuler l’efficacité de leur solution. L’inconvénient majeur de cette solution est que le nombre de voies possibles pour une intersection est limité par la portée de détection des capteurs. De plus, les mesures prises en compte sont basées minutes après minutes et non en fonction des phases ou cycle, qui restent fixes. Enfin, la définition des slots de temps n’est pas optimale : imaginons qu’un véhicule arrive avant le feu au moment où ce dernier passe au rouge, alors il ne sera pas pris en compte.

Comme remarqué, la plupart des auteurs s’occupent majoritairement de traiter le temps possible des feux verts. La sélection des phases reste (dans ce qui a été étudié) fixe comme l’illustre par exemple [84], où les auteurs se basent sur un ensemble de cas possibles, comme c’est le cas pour [96] ou encore [101]. Néanmoins, chaque auteur s’accorde sur l’amélioration de la fluidité du trafic routier, en comparaison



aux solutions classiques.

### Parenthèse : temps de feu vert

Utilisées par les systèmes traditionnels, nous détaillons ici quelques méthodes supplémentaires applicables aux capteurs, permettant d'obtenir un temps de feu vert adapté dans une approche dynamique, pour une phase donnée. Ce paramètre repose sur plusieurs aspects ([48]). Généralement, un temps de feu vert de passage  $T_p$  est utilisé afin de ne pas provoquer trop de cycles d'arrêts et de reprise. Il correspond au temps théoriquement nécessaire à servir les véhicules présents sur l'intersection. Soit au minimum  $T_p = T_s + T_h * N^{max}$ , avec  $T_s$  le temps de démarrage moyen d'une file estimé à 4 secondes,  $T_h$  le temps moyen de traversée du feu pour un véhicule estimé à 2 secondes, et  $N^{max}$  la taille en véhicules de la plus grande file d'attente. Notons que si un détecteur est uniquement placé au niveau du feu, alors  $T_p$  est au minimum de  $T_s + T_h = 6$  secondes, afin de permettre aux premiers véhicules d'arriver (généralement arrondi à 10 secondes pour plus de confort). Soit  $T_{max}$ , une borne maximale, si  $T_p < T_{max}$ , alors chaque véhicule supplémentaire ajoute un temps au feu vert, que nous pouvons estimer au minimum de  $h$  secondes, jusqu'à atteindre éventuellement  $T_{max}$ .

Le temps de feu vert maximum a fait l'objet de nombreuses études. Kell et Fullerton ([56]) observent que  $T_{max}$  doit se situer entre 30 et 60 secondes. Lin ([65]) étudie les relations entre  $T_{max}$  et le délai moyen d'attente, afin que  $T_{max}$  soit cohérent même aux heures de pointes. Orcutt ([60]) a suggéré que  $T_{max}$  devait être assez long pour laisser passer 1.3 fois la longueur moyenne de la file. Courage ([46]) a indiqué qu'un  $T_{max}$  élevé n'avait que peu d'impact sur un système adaptatif si le trafic est trop faible. Citons finalement [81] et [98] qui proposent des méthodes plus modernes afin de fixer un  $T_{max}$  cohérent, complexe à cerner de par la complexité et la diversité possible d'une intersection.

### Synthèse : Placement et nombre de capteurs

Le placement des capteurs sur une intersection peut facilement varier. Ici, nous allons tout d'abord justifier le nombre de capteurs qu'il est nécessaire mettre par voies, puis, nous verrons leur placement en détail.

**Nombre** Concernant le nombre de capteurs, il apparaît clair qu'une unité au minimum est nécessaire par voie afin de détecter avec cohérence l'ensemble des véhicules, ceci sans erreur et sans se borner à une infrastructure maximale (exemple : [102]). De plus, utiliser au moins un capteur par voie permet d'individualiser ces dernières et obtenir des données plus pointues. Ensuite, si le matériel est suffisant, il est judicieux de mettre en place deux capteurs par voie, tel que proposé précédemment, l'un au niveau du feu et l'autre à une distance  $d$  :

- Dans le cas de la théorie des files d'attente, le recueil des départs est une donnée facile à obtenir.
- Possibilité de détecter les véhicules fraudeurs (passage au feu rouge) et ainsi obtenir des statistiques exactes.
- Deux capteurs peuvent nous aider à obtenir plus de données : vitesse d'un véhicule, ou la longueur moyenne.
- Prévision du risque d'accident, détection d'inactivité, etc.
- Deux capteurs sont statistiquement plus efficaces qu'un seul capteur ([84]).

La communication entre les deux capteurs peut être imaginée sous différentes formes : nous pouvons imaginer la création d'une hiérarchie et l'utilisation d'une fonction d'agrégation entre les deux, ou plus simplement un envoi direct à la station de base. Il reste toutefois dommage de se limiter à une approche centralisée, plutôt que d'utiliser pleinement les capacités des capteurs, qui ne sont alors plus considérés comme des boucles électromagnétiques.

**Placement** Le calcul de la distance  $d$  à laquelle est placée le deuxième capteur est vue de deux manières dans la littérature.

D'une part, les **méthodes fixes** dont l'objectif est d'estimer une distance fixée afin de placer le deuxième capteur. Dans [96], les auteurs se basent sur les longueurs de queues moyennes effectuées dans [41] afin d'estimer la distance du capteur à huit véhicules, dans le cadre de leurs simulations. Les auteurs précisent tout de même que des ajustements doivent être faits en fonction des conditions de trafic, et tout autre facteur n'entrant pas en compte dans le modèle théorique. Dans [84], après plusieurs simulations, il est constaté que le deuxième capteur doit se situer entre cinq et huit véhicules du feu, sous peine de provoquer des erreurs de mesures (exemple : un véhicule changeant de voie). De plus, il est constaté qu'une distance supérieure à deux voitures n'a aucun effet si le trafic n'est pas chargé. Cette méthode va permettre d'adopter un schéma similaire pour toutes les intersections, mais ne va pas prendre le cas où le trafic est trop important : si l'ensemble des intervalles de capteurs sont saturés, le système est rendu inefficace.

D'autre part, les **méthodes dynamiques**, qui vont prendre en compte le trafic et résoudre le problème des méthodes fixes tout en nécessitant davantage d'organisation dans la mise en place. Dans [101], les auteurs estiment la position du capteur être à  $d = v * g^{max}$ , avec  $v$  la vitesse moyenne constatée. Exemple : en admettant que les automobilistes aillent à  $\sim 20\text{km/h}$  à l'approche d'un feu rouge ( $\sim 7\text{m/s}$ ) et que le temps de feu maximum soit de 10 secondes, le capteur serait situé à 70m, permettant ainsi de mesurer l'ensemble de la file (ce qui est intéressant uniquement si celle-ci est en permanence pleine, et donc que le débit moyen du trafic dépasse une certaine valeur). Dans [86], il est décrit une méthode dynamique étant appliquée plus classiquement par les ingénieurs des ponts et chaussées. La distance  $d$  (mètres) des deuxièmes capteurs est déterminé à l'aide d'une formule se basant sur un temps de feu vert minimum de la voie concernée, avec  $L_v$  la longueur d'un véhicule estimée à 6 mètres en moyenne :

$$d = (g_{min} - t_s) \cdot \frac{L_v}{h}$$

Au final, si les méthodes fixes nous permettent d'estimer le placement d'un capteur quel que soit l'état du trafic routier, il nous faut garder en tête que ce dernier va jouer un rôle essentiel : un capteur placé à une distance fixe n'aura pas le même effet sur une intersection constamment embouteillée que sur une intersection très fluide. Le placement des capteurs, au même titre que les détecteurs en général, est une opération délicate car dépend également de l'infrastructure routière, ainsi que des connectivités (exemple : distance de communication en sans-fil).

**Parenthèse : calcul de AJWT/ATWT** Comme introduit dans la première partie, l'une des métriques les plus couramment utilisée dans la littérature est le temps moyen d'attente pour un voyage complet (ATWT), calculé à partir du temps moyen d'attente à une intersection (AJWT). Nous nous intéressons ici à son calcul, abordé dans la littérature. [84] l'illustre par le biais du simulateur GLD (*voir plus bas*). Ainsi, AJWT au cycle  $i$  peut être donné par :

$$AJWT_i = \frac{AJWT_{i-1} \cdot (\Phi - 1) + \Gamma}{\Phi}$$

Où  $\Phi$  représente le nombre de véhicules entrant et  $\Gamma$  une fonction de délai représentant le retard accumulé des véhicules avant d'arriver au feu. Le temps moyen d'attente pour un voyage complet peut lui être représenté par :

$$ATWT = \frac{\sum_{j=0}^M AJWT_j \cdot \Phi}{\sum_{j=0}^M \Phi_j}$$

En considérant une infrastructure constituée uniquement d'intersections et où les véhicules arriveraient et iraient vers des nœuds de bordures (*cas GLD plus bas*),  $M$  représenterait le nombre total de ces nœuds. Les deux fonctions précédentes peuvent être utilisées afin d'évaluer les performances d'un contrôleur de feux.

## Synthèse : comparaison des modèles

Les modèles abordés précédemment peuvent être comparés au travers du tableau 4. Comme indiqué plus haut, ici nous ne souhaitons pas comparer les résultats bruts de chacune des solutions, nous comparons donc leurs caractéristiques.

Comme constaté, aucun modèle ne répond parfaitement à l'ensemble des contraintes que peut regrouper la problématique de la gestion des feux. De plus, un ensemble commun d'inconvénients peut être levé : utilisation des réseaux de capteurs sans fil peu formelle (gestion énergétique, distance et fréquence de communication), calculs s'ajustant en fonction du trafic mais pré-déterminés à un cycle d'avance (manque de temps-réel) ou encore manque de souplesse dans la constitution du plan de feux. Finalement, chacun utilise son modèle et détaille un aspect sans aborder l'ensemble de la problématique. Cette dispersion opère également sur les mesures de performances : les simulateurs ne se ressemblent pas, les métriques portent un nom similaire mais n'ont pas le même point de mesure, ce qui nous empêche d'effectuer des comparaisons de performances.

-	Nombre de capteurs	Position	Distance du deuxième capteur	Gestion des phases	Gestion du temps aux feux	Gestion énergétique des capteurs	Facteurs pris en compte	Remarques	Simulateur
Yousef et al. ([96])	2 / voie	Au niveau du feu, et avant le feu	Fixe	Dynamique	Dynamique	Prise en compte	Nombre de véhicules	-	Simulateur interne
Tubaishat et al. ([84])	2 / voie	Idem	Aucun modèle de défini	Semi-dynamique	Non défini, sinon irréaliste	Non détaillée	Nombre de véhicules	-	GLD
Zhou et al. ([101])	2 / voie	Idem	Dynamique	Semi-dynamique	Dynamique	Non détaillée	Multiplés (par priorités)	Hypothèses lourdes	iSensNet
Zou et al. ([102])	1 / direction	Bordure de route, avant le feu	-	-	Logique floue	Prise en compte	Nombre de véhicules	Infrastructure limitée	Sur le terrain / calculs

Efficacité :

Mauvaise	Moyenne	Non optimale	Bonne
----------	---------	--------------	-------

TABLE 4 – Comparaison des modèles présentés en 3.3.1

Avant de conclure, nous étudions brièvement le cas d'architectures plus complexes.

### 3.3.2 Sur plusieurs intersections

La logique de conception étant distribuée, il est facile de reprendre le modèle général précédemment étudié et de le généraliser sur plusieurs intersections. Les données prises en compte seront donc similaires pour chacune des intersections, et d'autres mesures viendront s'ajouter en provenance des intersections voisines. Par exemple, [96, 101] étendent leur modèle à plusieurs intersections respectivement dans [96] et [100]. Généralement, le but de tels modèles est de créer une "vague verte", permettant à un flux d'usagers de traverser un ensemble d'intersections en minimisant au maximum les arrêts, ceci nécessitant une coordination particulière des temps de feux entre les contrôleurs. Le gros inconvénient de ces modèles est que la communication - souvent considérée comme directe entre deux intersections - n'est pas nécessairement réaliste : exemple avec un réseau purement sans fil et deux intersections voisines très éloignées l'une de l'autre. De plus, de tels modèles mettent en scène une infrastructure souvent simple (réseaux maillés la plupart du temps, comme l'illustre la figure 10) et ne prennent pas en compte des éléments essentiels pourtant existants entre deux intersections : stops, priorités ou encore feux intermédiaires. Malgré ceci, il est évident que les réseaux de capteurs sans fil possèdent les capacités nécessaires à l'accomplissement d'un tel système distribué, sous réserve d'un débit de données suffisamment grand et d'un partage équitable du spectre radio.

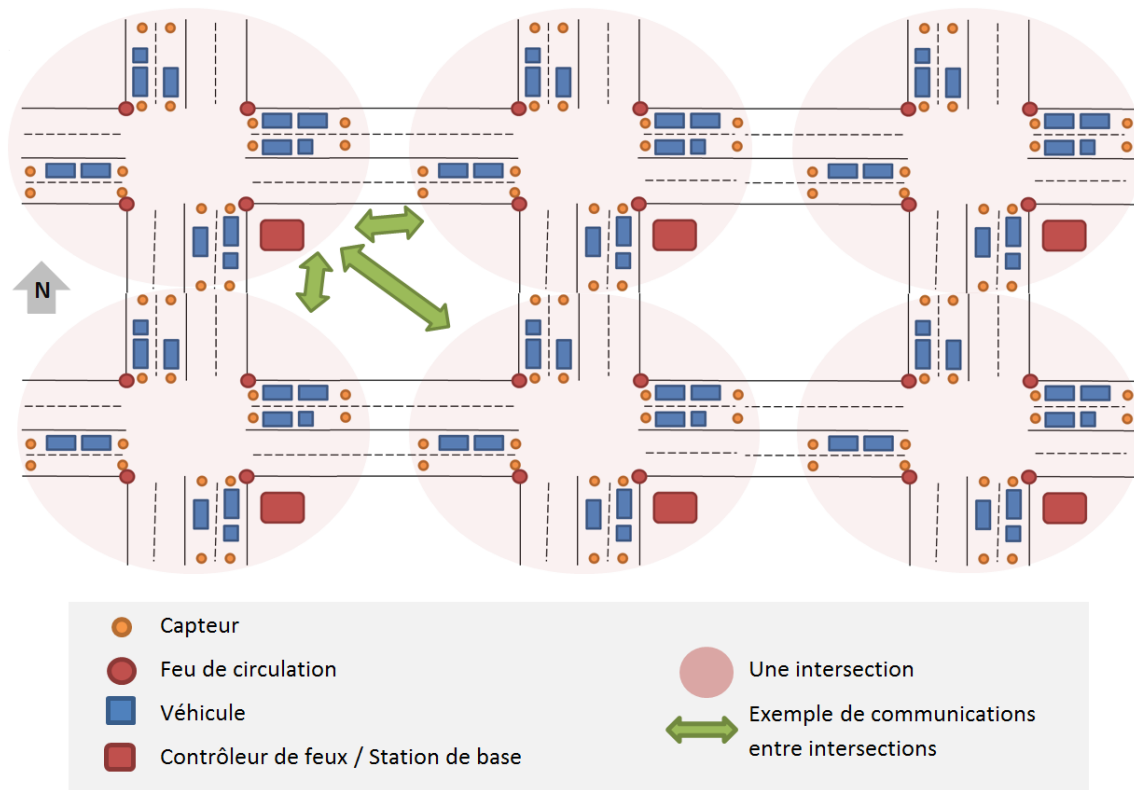


FIGURE 10 – Schéma général d'un modèle se reposant sur plusieurs intersections

### 3.3.3 Applications particulières

Outre les intersections, les réseaux de capteurs sans fil vont servir à de nombreuses applications, tels que la sécurité.

Barnes et al. ([30]) utilisent le filtre de Kalman, qui leur permet d'estimer le risque de collision à une intersection entre deux véhicules en fonction de paramètres transmis par les capteurs électromagnétiques (détection ou même vitesse) : prévision de l'évolution probable du système. Par dessus ce système, nous pourrions imaginer une implantation de PMV afin d'informer l'utilisateur des risques encourus.

Liao et al. ([64]) décrivent une solution de monitoring du trafic dans une résidence : des capteurs sont placés à chaque entrée et à chaque intersection, ceux-ci permettant d'avertir les utilisateurs en cas de collision possible ou de trop grande vitesse. Un aspect intéressant que nous pourrions étendre aux parkings.

Enfin, nous pouvons citer l'utilisation des capteurs sans fil dans des modèles coopératifs, où des systèmes embarqués à bord de véhicules sont introduits au réseau. Les possibilités sont ici décuplées, comme le montre notre partie 2.5.3 où nous avons notamment abordé le cas des contributions [75, 89, 90, 39] utilisant de tels réseaux comme base à leurs modèles. Ce document étant essentiellement ciblé sur la description de réseaux fixes, nous ne détaillerons pas davantage cet aspect de la littérature.

### 3.3.4 Evolution du système actuel

Afin de voir plus loin, les auteurs de [88] ont identifié deux possibilités afin d'implanter et utiliser au mieux les réseaux de capteurs sans fil à l'échelle d'une grande zone urbaine. En premier lieu, un système

entièrement ad-hoc tel que décrit précédemment où chaque élément aurait la possibilité de communiquer avec ses voisins et en multi-sauts avec l'intégralité du réseau. Un tel système incorporerait des éléments pouvant communiquer à des distances variables, et aurait une logique de fonctionnement entièrement distribuée. Les avantages en termes de réactivité et performances sont évidents, mais restent très difficile à mettre en place face aux systèmes traditionnels massivement implantés et financés. En deuxième lieu, les auteurs de [88] et particulièrement [95, 94] évoquent une architecture hybride, où les systèmes actuels resteraient présents, tout en permettant l'intégration progressive de systèmes distribués. Ces derniers seraient rassemblés dans une couche de bas niveau qui serait gérée par un centre de contrôle : ceci permettrait d'économiser les coûts et les travaux de mise en place tout en bénéficiant de cette technologie, ce qui représente la solution la plus plausible actuellement. Ceci peut être illustré au travers de la figure 11.

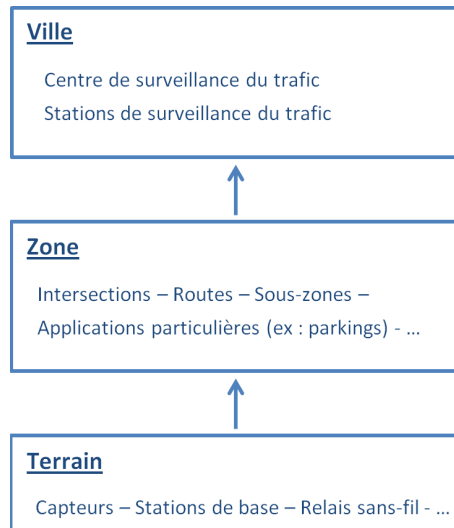


FIGURE 11 – Architecture hybride : systèmes actuels couplés aux réseaux de capteurs

### 3.3.5 Conclusion

La technologie des réseaux de capteurs s'est beaucoup développée durant ces dernières années, et son impact sur les réseaux routiers urbains peut être d'une grande importance. Si de nombreux auteurs se sont penchés sur le couplage de tels réseaux avec des technologies de gestion de trafic plus classiques, les résultats restent tout de même partiels et dispersés : si de nombreux aspects sont présents, aucun n'arrive réellement à synthétiser l'ensemble des caractéristiques qu'il importe de connaître pour réguler au mieux le trafic routier urbain et utiliser les réseaux de capteurs sans fil, technologie particulière. De plus, l'ensemble reste très peu généralisé : prenons le cas d'une intersection où sortir du modèle représenté par la figure 8 n'est pas nécessairement aisé, surtout dans le cas d'intersections plus complexes. Ensuite, nous regrettons l'absence d'extensions de chacun des modèles : lorsque la gestion des feux d'une intersection est modélisée et éventuellement étendue à plusieurs, il est rare de retrouver par dessus ces modèles des systèmes prenant en compte les voies spéciales, les passages piétons (exemple : [29]) ou encore la prévention des accidents. Des éléments pourtant essentiels dans une situation réelle. Le cas particulier des capteurs est, quant à lui, souvent évasif : des notions essentielles telles que la fréquence de communication, la relation avec la distance ou encore les cycles d'éveil et de sommeil sont peu voire pas abordés dans le cas d'une simple intersection. Ce constat est d'autant plus flagrant à l'échelle de plusieurs intersections où les modèles proposés manquent cruellement de réalisme. Malgré tout ceci, nous ne pouvons que souligner la réactivité et la souplesse des réseaux de capteurs sans fil, qui possèdent un avenir certain dans le domaine des systèmes de transport intelligents (*exemple : 2.5.4.*).

## 4 Simulations

L'utilisation d'outils afin de modéliser et simuler le trafic routier est courant. Ci-dessous, nous détaillons quelques solutions en s'orientant vers plusieurs objectifs : l'existence des paramètres et métriques présentées dans la partie 1., la possibilité d'implémenter des algorithmes de contrôle en se reposant sur des détecteurs, de constituer des cartographies réalistes et complètes, et enfin une application libre.

### 4.1 Présentation

Afin de modéliser le trafic routier, et particulièrement les intersections et les systèmes de changement de feux, de nombreuses solutions existent : tandis que les auteurs de [73] effectuent un bref comparatif des simulateurs touchant au trafic routier (qu'il soit urbain, concerne les transports de marchandises, ou autres), les auteurs de [45] listent quelques outils essentiellement commerciaux permettant de simuler le cas des intersections et abordent également le cas de simulations effectuées depuis des logiciels de calcul numérique tels que MatLab ou Scilab (équivalence gratuite), utilisés pour calculer des valeurs telles que décrites plus haut (3.3.1 / *Synthèse : données utilisées et simulations*). Enfin, nous pouvons citer le cas de certains auteurs qui choisissent de construire leur propre modèle réel ([96] ou encore [34] avec la plateforme iSensNet). Dans cette partie, nous avons choisi pour notre part de développer le cas de deux solutions existantes et nous ayant séduites : GLD (*Green Light District Simulator*) et SUMO (*Simulation of Urban MObility*). Ces solutions possèdent l'avantage d'être sous licence GPL et semblent bien adaptées à ce que nous recherchons, à savoir un simulateur capable de modéliser des situations réalistes et parfois complexes, tout en nous donnant la possibilité d'agir en implantant certains éléments (algorithmes, détecteurs, etc.).

#### 4.1.1 Green Light District Simulator - GLD

Le logiciel open-source GLD ([93]) est un simulateur microscopique<sup>1</sup> de gestion du trafic routier à temps discret et espace continu entièrement écrit en Java par Wiering et al. (téléchargement du projet : [10, 11]). GLD permet d'évaluer les performances des algorithmes de contrôleurs de feux mais également de navigation et d'apprentissage au sein de systèmes multi-agents. Les infrastructures sur lesquelles il est possible de se baser sont composées essentiellement de nœuds d'entrées et de sorties de véhicules (nœuds de bordures) et de nœuds de carrefour (avec ou sans feux de circulation). Ce logiciel, dont la dernière version date de 2006<sup>2</sup>, se compose d'un éditeur d'infrastructures ainsi que d'un simulateur, et a l'avantage d'être très bien documenté. Quelques auteurs l'utilisent dans la littérature, tels que [85] ou encore [92]. Une capture d'écran du logiciel est présente à la figure 12.

**La partie éditeur** D'un côté, la partie éditeur va nous permettre de créer une infrastructure, notamment en nous donnant la possibilité de placer des nœuds (carrefours ou bordures par exemple), des routes permettant d'effectuer la liaison entre ces éléments, des voies constituant ces routes (nombre, possibilités de tourner à gauche ou droite, etc.).

**La partie simulateur** D'un autre côté, la partie simulateur nous permet d'effectuer des simulations sur les différentes infrastructures mises en place. Tout d'abord, nous pouvons mesurer et tracer l'évolution d'un ensemble de valeurs, utilisées comme comparatif dans une majorité de la littérature, par exemple : le temps d'attente moyen à un feu ou sur le parcours, la taille moyenne actuelle de la file d'attente ou encore le nombre d'utilisateurs arrivés depuis le début de la simulation. Ensuite, nous pouvons changer un ensemble de paramètres, tels que le comportement d'un conducteur, la fréquence d'apparition d'un nouveau véhicule depuis un nœud d'entrée, la vitesse d'exécution du programme, et surtout, l'algorithme du contrôleur de feux. Cette dernière option représente une fonction très intéressante pour nous : le logiciel

1. Dans le cadre de la gestion du trafic routier, un simulateur microscopique va modéliser le comportement individuel des véhicules, contrairement à l'approche macroscopique qui va modéliser le trafic sous forme de flux (à un plus haut niveau).

2. En date de novembre 2011.

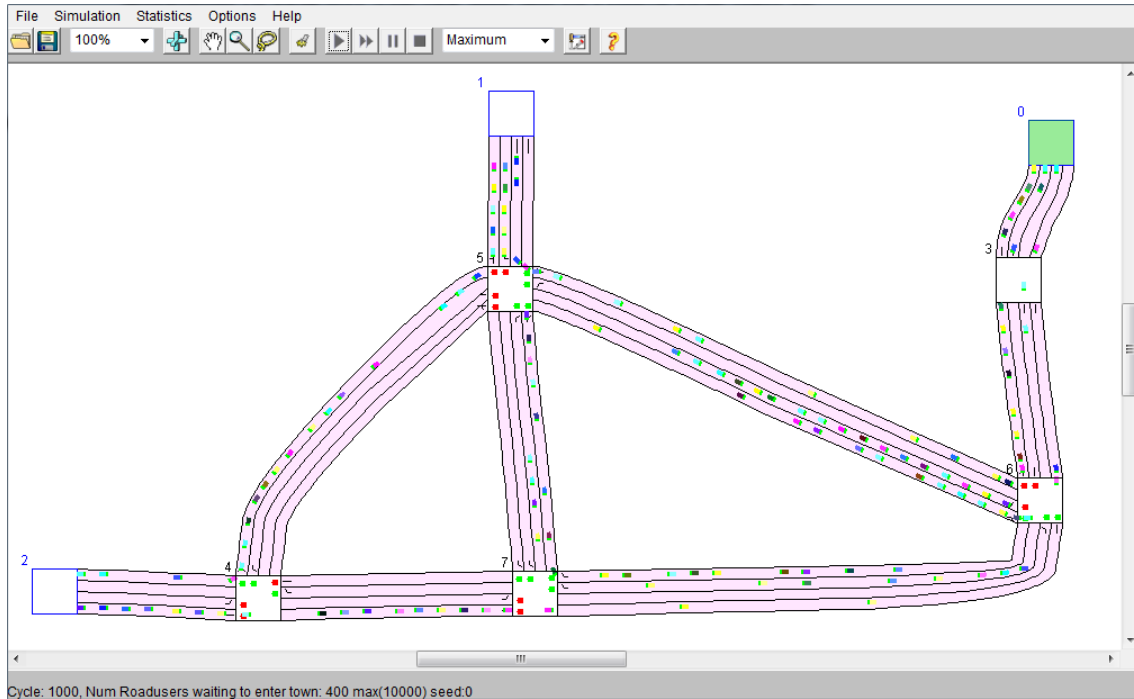


FIGURE 12 – Green Light District Simulator

propose un ensemble d’algorithmes pré-implantés de gestion des feux, et son architecture logicielle nous permet d’en mettre en place de nouveaux très facilement. Ces algorithmes peuvent être de deux types : basés sur l’infrastructure (sur les données du trafic), ou basés sur les véhicules (agents, l’objectif est alors de maximiser le gain individuellement). Le type nous intéressant ici est le premier, qui utilise un système de gains : plus une voie ou une combinaison possible de voies ont de gains, plus leur usage sera probable. Comparons ici cinq exemples d’algorithmes du premier type mis en place dans le logiciel, et étant du premier type :

- **Random** : algorithme de base, avec un temps fixe égal pour toutes directions, en comparaison à la technologie "fixe" bien souvent encore en place.
- **Most cars** : toutes les voies qui ont au moins un véhicule ont une chance de passer.
- **Best first** : s’occupe des configurations de voies ayant le plus grand nombre de véhicules en premier afin de les soulager.
- **Longest queue** : accorde le feu à la voie qui a la plus grande queue, et non à la combinaison.
- **Relative longest queue** : division du nombre d’utilisateurs sur la longueur de la voie en elle-même, afin d’avoir une information relative au carrefour.

La figure 13 effectue un comparatif de ces algorithmes, rapide puisque illustré au travers d’une simulation de 5000 cycles chacun. Le cycle étant l’unité de temps de GLD, et correspondant à un pas logiciel (un déplacement). Les simulations ont été effectuées sur une infrastructure composée de quatre carrefours, avec pour probabilité d’entrée des nœuds de 0.4 (par cycle, pour chaque nœud de bordure).

Notons que la toute dernière version de GLD prend en compte des paramètres intéressants comme la possibilité de définir une probabilité d’accident à chaque cycle et de voir les répercussions sur le temps d’attente moyen, de définir une probabilité dynamique d’arrivée des véhicules, ou encore de borner le nombre de véhicules en attente d’entrer dans le parcours. Notons également la présence d’un logiciel basé sur GLD et dont la dernière version date de 2008 : MoreVTS ([12]). Développé par I-Atracos, ce logiciel open-source ajoute notamment à GLD une interface avec des systèmes externes (envoi de configurations sur des ports, et prise en compte par le logiciel).

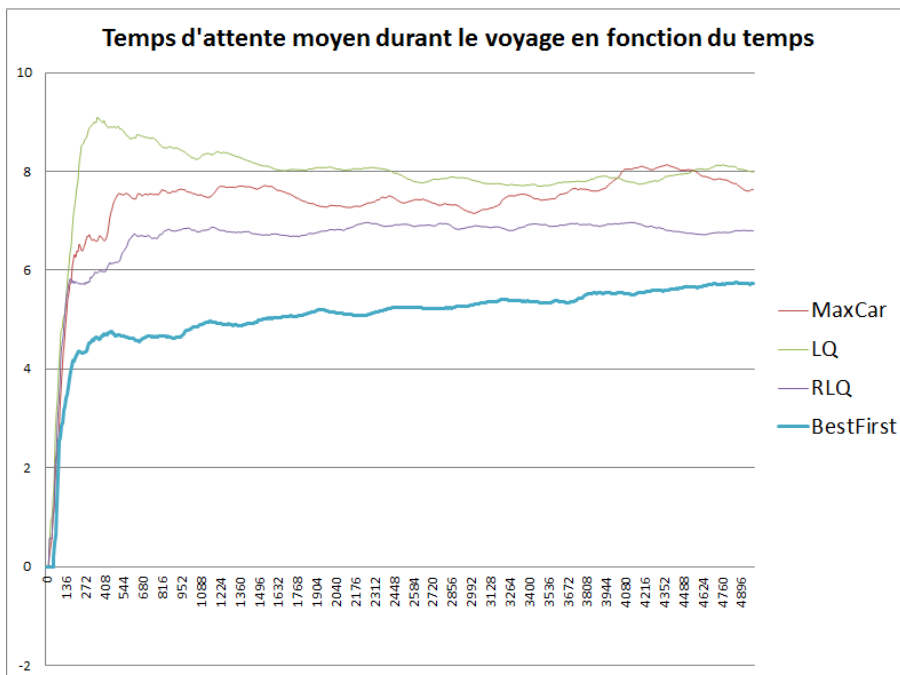
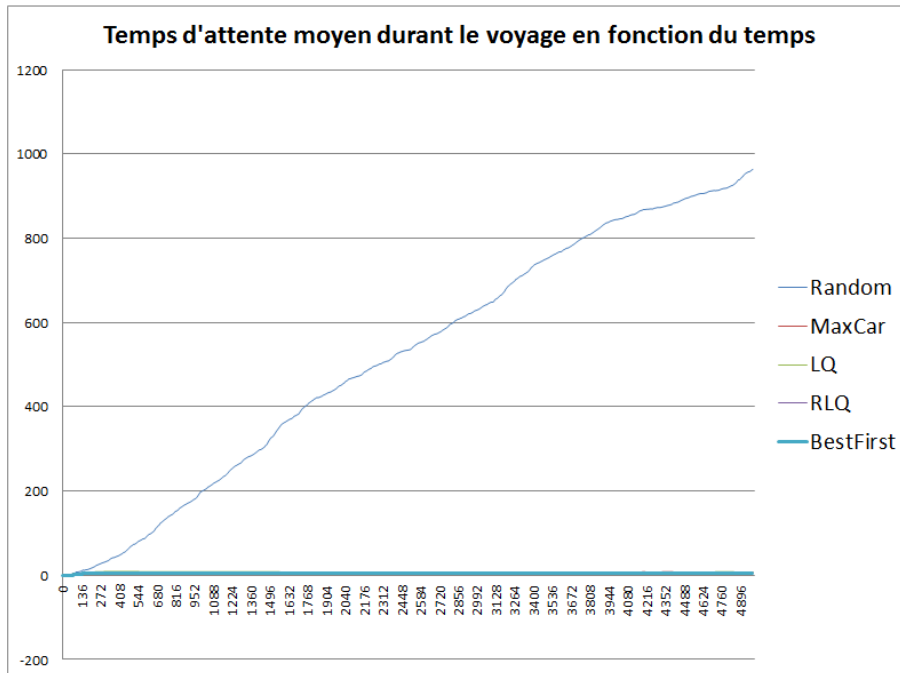
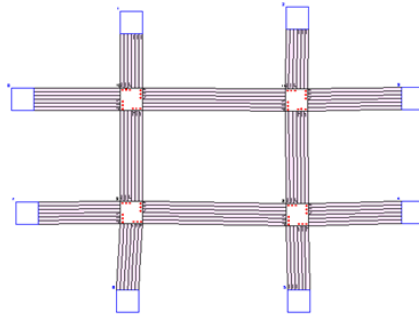


FIGURE 13 – Comparatif de quelques algorithmes



Pour finir, nous pouvons aborder les limites du logiciel. Tout d'abord, selon Marco Wiering, GLD ne prend pas en compte par défaut le système traditionnel de phases et de cycles, ce logiciel étant à la base pensé à des fins d'optimisation d'algorithmes, essentiellement pour des cas de systèmes multi-agents. De plus, le plan de feux (constitué uniquement du feu vert et rouge) est établi dynamiquement et est régénéré à chaque pas, ce qui ne représente pas une situation réaliste. Malgré ceci, GLD s'avère être un excellent simulateur, très souple de par l'utilisation de Java comme langage de mise en place, et étant conçu pour l'exécution d'algorithmes de contrôleur de feux. Les limites que nous venons de décrire étant programmables assez facilement.

#### 4.1.2 Simulation of Urban MObility - SUMO

SUMO ([31]) est un simulateur open-source à temps discret, espace continu et microscopique entièrement réalisé en C++ permettant de modéliser l'écoulement du trafic routier, ceci en traitant une gamme de cas plus réalistes et plus larges que GLD (qui se limite principalement à l'optimisation du temps de parcours). Bien que plus complexe à mettre en place, SUMO possède également l'avantage d'être toujours maintenu par ses auteurs et d'être doté d'une documentation complète. La dernière version du logiciel date de novembre 2011<sup>1</sup> et est disponible sur la page officielle du projet ([13]). Globalement, l'utilisation de ce logiciel peut se décomposer en trois points détaillés ci-après.

**La construction d'un réseau** Contrairement à GLD, la génération d'une carte (ici appelé un réseau) n'est pas nécessairement automatisée par une interface mais repose sur la construction d'un fichier XML de type `projet.net.xml`, qui peut être obtenu de multiples manières en fonction de la volonté de mise en place. Cet ensemble de possibilités peut paraître déroutant pour un utilisateur non averti, mais reste très puissant : là où GLD proposait la mise en place d'une infrastructure simple, SUMO peut mettre en place des structures plus complexes, incorporant le changement de voie, des boucles électromagnétiques, la création de types de véhicules, et bien d'autres. Deux principaux outils existent pour générer de tels réseaux :

- **NETGEN** permet de générer en ligne de commande un réseau de plusieurs formes : en grille, en toile ou aléatoire.
- **NETCONVERT** possède la capacité de générer des cartographies adaptées pour SUMO à partir de différents formats.
  - De base, il est possible de définir un ensemble de fichiers XML afin d'obtenir le réseau souhaité. Cette démarche peut sembler lourde à faire manuellement, mais permet de mieux cerner le fonctionnement du logiciel, et reste suffisante pour de petits réseaux. Un exemple de mise en place est présent en annexe, et représente l'intersection détaillée sur la figure 8.
  - Tout d'abord, il est nécessaire de décomposer la carte souhaitée en un ensemble de nœuds, qui peuvent être de plusieurs types : de bordures, fictifs pour la réduction ou l'augmentation de voies, ou encore d'intersections, avec ou sans feux (règle de la priorité à droite). Ces nœuds et leurs coordonnées respectives prennent place dans un fichier de la forme `projet.nod.xml`.
  - Ensuite, un fichier `projet.typ.xml` permet de mettre en place les types de routes souhaitées, incluant pour chaque type des informations telles que le nombre de voies ou encore la vitesse limite. Ce fichier de types est optionnel, les paramètres pouvant directement être indiquées lors de la construction des routes.
  - Puis, un fichier `projet.edg.xml` permet de relier les nœuds par des routes.
  - Le fichier `projet.con.xml` permet pour sa part de mettre en place les directions possibles pour chaque voie.
  - Le fichier `projet.netc.cfg` permet d'indiquer tous ces fichiers en vue de la création de la carte par le programme **NETCONVERT**, permettant d'obtenir le fichier `projet.net.xml`.
- Pour plus de confort, **NETCONVERT** possède l'avantage de prendre en charge de nombreux types de cartes existants, en provenance de certains simulateurs ou plateformes reconnues, telles que MatSim ([14]) ou encore OpenDrive ([15]). Particulièrement, OpenStreetMap ([16]) est également pris en charge : il s'agit d'une carte mondiale en partage et totalement libre, dont chacun peut contribuer à la construction ainsi qu'exporter des données, généralement au format `.OSM`.

---

1. En date de novembre 2011.

Cette extension peut être lue par des logiciels tels que JOSM ([17], qui peut également servir à créer des cartes. Le logiciel eWorld ([18]) peut également, en plus de lire directement des configurations SUMO, faire office d'éditeur de fichiers `.OSM` ou importer des données en ligne. Finalement, il est possible d'obtenir un fichier de type `projet.net.xml` à partir de formats `.OSM` via `NETCONVERT`. Plus loin encore, certaines structures décoratives telles que les rivières, les immeubles et autres éléments graphiques peuvent être importés à l'aide de `POLYCONVERT`, également inclus dans SUMO. Un exemple de création peut être illustré au travers de la figure 14 avec un projet interne à SUMO : *TAPASCologne*, qui vise à reproduire la ville de Cologne (Allemagne), aussi bien au niveau de la cartographie que du trafic routier en lui-même.

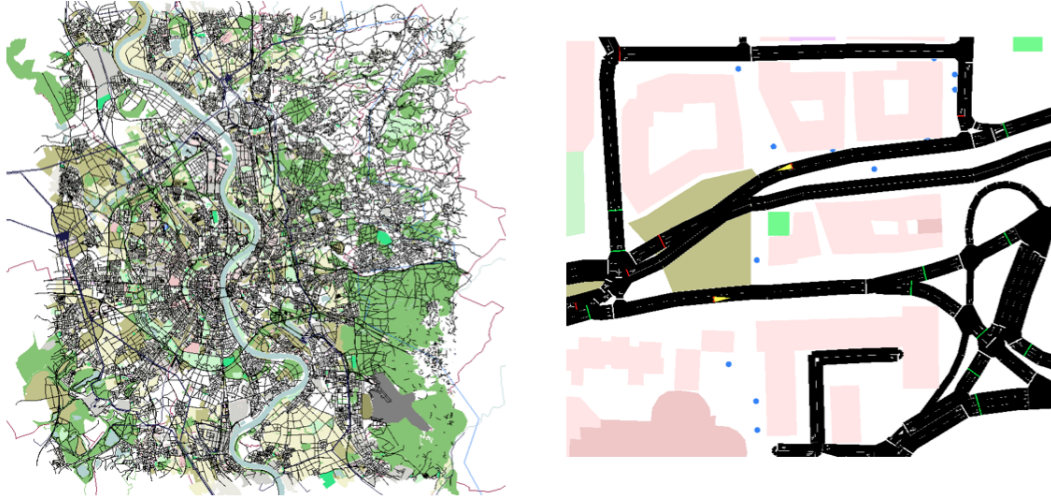


FIGURE 14 – SUMO - Conversion de fichiers `.OSM`

**Modélisation de la demande** Nous venons de voir que la génération d'un réseau routier pouvait se faire par plusieurs moyens : soit à la main directement, soit via les utilitaires fournis avec le simulateur. Une fois ce réseau obtenu, l'étape suivante consiste à définir les informations à propos des véhicules, et notamment des routes qu'il est possible pour eux d'emprunter.

**Définition 11.** Un voyage est un déplacement d'un véhicule d'un endroit à un autre, en ayant pour information le lieu de départ et d'arrivée.

**Définition 12.** Une route est un voyage étendu où l'ensemble des endroits dans lequel le véhicule passe est connu.

SUMO a besoin de connaître les routes possibles du réseau afin de générer les mouvements des différents véhicules. Le moyen le plus simple d'en obtenir est de les éditer manuellement, mais uniquement si leur nombre n'est pas trop élevé. La démarche manuelle va donc être de créer soi-même un fichier du type `projet.rou.xml` qui va se charger de définir les types de véhicules existant ainsi que les routes possibles. Les véhicules en circulation sur la simulation vont être alors indiqués en utilisant ces deux paramètres, et de deux manières : soit sous forme de flots arrivant de manière continue pendant une certaine période, soit individuellement. Sur de larges réseaux, SUMO fournit plusieurs applications permettant de générer ce fichier en fonction de multiples critères :

- `DUAROUTER` : sur la base d'un ensemble de voyages, cette application obtient les routes correspondantes, en cherchant le plus court chemin entre le point de départ et d'arrivée suivant l'algorithme de Dijkstra.
- `DJTRROUTER` reprend des principes similaires, mais utilise des probabilités de tourner afin de générer les routes.

- DFROUTER s'appuie sur les mesures effectuées par les détecteurs présents.
- ACTIVITYGEN se repose sur une description de la population vivant sur le réseau : nombre d'habitants, de ménages, la probabilité pour un adulte d'avoir une voiture, d'utiliser les transports en commun, description des zones d'activité de la ville (ex : école), etc.
- Notons la possibilité de générer des routes aléatoirement en s'aidant d'un outil inclus dans SUMO et des deux premières applications.

**Simulations** Une fois le réseau et les informations du trafic souhaité générées, il faut mettre en place le fichier de configuration correspondant à la simulation, de la forme `projet.sumo.cfg`. Ce fichier va rassembler les deux fichiers `projet.net.xml` et `projet.rou.xml`, ainsi que d'autres éléments, tels que :

- Des éventuels fichiers supplémentaires (pour la définition de stops, de détecteurs, etc.).
- Le temps de simulation (début et fin, en pas de programme, ajustables en millisecondes). Exemple : je veux une simulation entre le pas 1000 et le pas 1500.
- Les valeurs à enregistrer : le programme générera alors en fin de simulation un ou plusieurs fichiers XML contenant les résultats. A ce propos, de très nombreuses valeurs sont exploitables afin de pouvoir évaluer au mieux un modèle, comme par exemple les temps de parcours pour chaque véhicule étant entré dans le réseau au moment de la simulation.

La simulation ainsi définie est exécutable par l'intermédiaire de deux programmes : via une interface graphique (SUMO-GUI, figure 15) mais également directement en ligne de commande (SUMO), où il est possible de passer certains arguments habituellement définis dans le fichier de configuration directement dans la commande. L'affichage de l'interface graphique apparaît propre, et de divers types : très basique, standard ou sous une vue plus réaliste. La possibilité de régler la vitesse d'exécution de la simulation existe toujours, ceci est ajustable en définissant le nombre de millisecondes auxquelles correspond un pas de programme. Les paramètres d'affichage sont entièrement personnalisables : couleur des véhicules en fonction de leur vitesse, de leur provenance, etc.

Concernant le cas des feux de circulation, ces derniers possèdent ici les trois couleurs (rouge, orange, vert) et suivent un principe de cycle et de phase clairement définis et visibles (contrairement à GLD où l'enchaînement allait à la performance et manquait de réalisme). Par défaut, NETCONVERT et NETGEN s'occupent de générer les feux de circulation et leur fonctionnement lors de la mise en place du réseau. Seulement, ces fonctionnements ne suivent pas nécessairement ce que nous pouvons retrouver dans la réalité. Ainsi, il est possible de charger des programmes supplémentaires afin de régir le fonctionnement des feux de circulation : l'utilisateur peut configurer à sa convenance le système de gestion des feux ([59]), l'opération se faisant par défaut manuellement avec un fichier de configuration fixe. Enfin, des détecteurs peuvent être mis en place dans des fichiers additionnels (manuellement, ou via des utilitaires intégrés). Ces derniers peuvent servir au recueil de données et à l'évaluation des performances de feux de circulation.

Dans ce document, nous ne pouvons évidemment pas décrire l'ensemble des fonctionnalités du programme, mais juste insister sur le fait que le logiciel possède un ensemble d'outils très large : dernier exemple avec le cas de TraCI (*Traffic Control Interface*). L'idée de base de cet outil est de donner l'accès à une simulation en cours, afin de récupérer des données en temps réel et éventuellement de modifier certains paramètres. Cet utilitaire va être particulièrement adapté à nos besoins. En tout, trois classes de possibilités :

- **Action sur la simulation et l'interaction client/serveur** : clore la connexion, effectuer un pas de programme, calculer certaines valeurs, etc.
- **Obtention d'informations** : en provenance des détecteurs, des feux de circulation, des voies, routes, véhicules, routes ou même sur la simulation en elle-même.
- **Modification d'état** : des feux de circulation, des voies ou encore des véhicules.

Terminons par décrire brièvement l'ouverture dont fait preuve ce simulateur, en permettant par exemple à TraCI de s'interfacer avec des logiciels de tout type : solutions intégrées (TraCI en Python, TraCI4J en Java) ou simulateurs réseaux répandus (Middlewares Trans ou MOVE pour NS2, iTETRIS pour

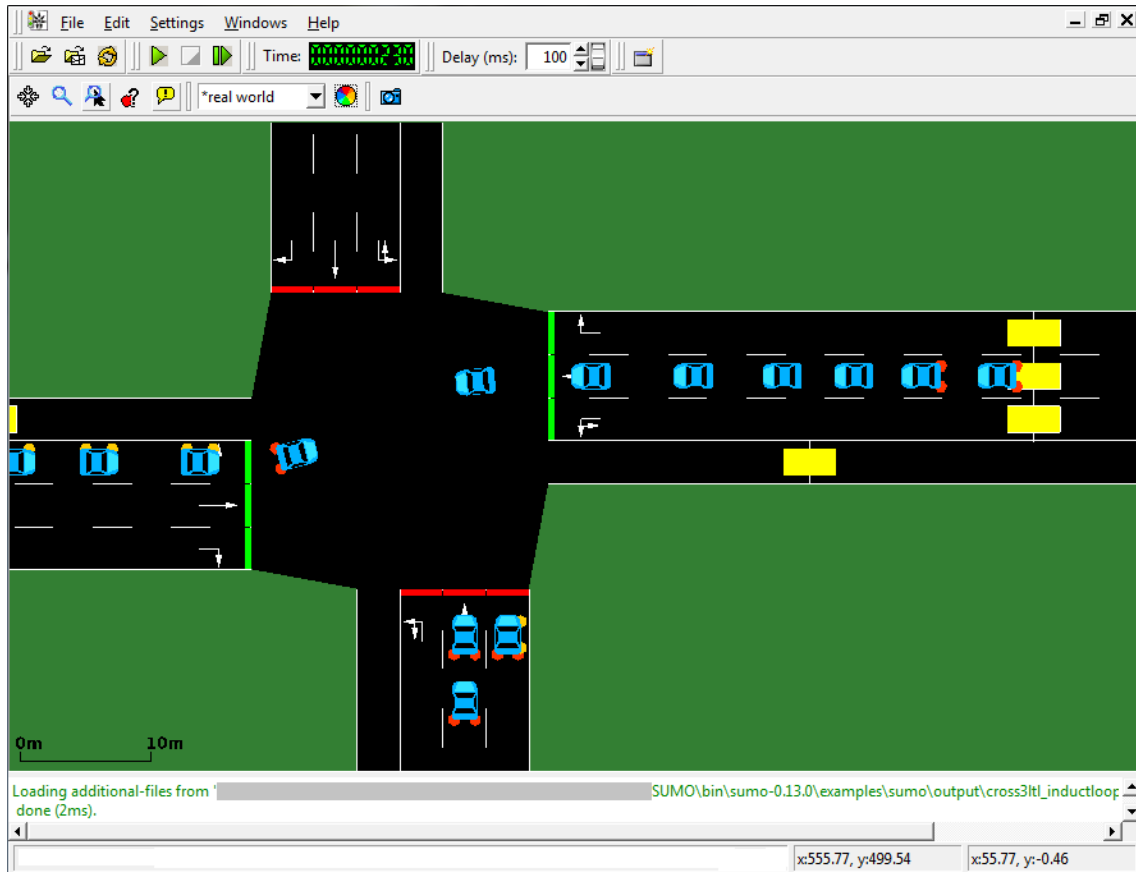


FIGURE 15 – SUMO

NS3, VEINS ou VSIMRTI pour OMNet++). De plus, les outils tiers sont nombreux, rendant l'utilisation du simulateur agréable, passée une phase d'apprentissage un peu difficile.

### 4.1.3 Conclusion

Nous avons présenté deux simulateurs, idéaux dans le cadre de nos études sur la gestion du trafic routier et notamment du cas des algorithmes de contrôleurs de feux. Après comparaison (voir tableau 5), il en ressort un net avantage pour SUMO, qui est beaucoup plus réaliste et riche que GLD, offre la possibilité d'utiliser des détecteurs (idéal dans le cas des réseaux de capteurs) et toujours maintenu. Toutefois, l'un des aspects négatifs du logiciel est la complexité de mise en place d'une simulation, qui est beaucoup moins évidente que sur GLD. Ce dernier restant un logiciel performant, plus intuitif et orienté vers l'optimisation de la gestion des feux de circulation : un bon complément qui pourrait s'avérer intéressant dans le cas de systèmes multi-agents ou de comparatifs entre plusieurs algorithmes de gestion des feux.

-	Type	Lang.	Temps / Espace	GPL	Dernière version <sup>1</sup>	Orientation	Maturité et possibilités	Niveau de réalisme	Connaissances en développement	Orientation de l'interface
GLD	Simulateur microscopique	Java	Discret / Continu	Oui	Jan. 2006	Optimisation du temps de parcours	Moyennes	Moyen	Petites	Configuration, visualisation et résultats
SUMO	Simulateur microscopique	C++	Discret / Continu	Oui	Nov. 2011	Multiple	Elevées	Elevé	Moyennes	Visualisation

Convenance :

Mauvaise	Moyenne	Non optimale	Bonne
----------	---------	--------------	-------

TABLE 5 – Modèles des deux simulateurs SUMO et GLD  
<sup>1</sup> En date de novembre 2011

## 5 Conclusion et perspectives

Au fil de ce document, nous avons vu plusieurs aspects de la gestion du trafic routier urbain. Nous avons commencé par une présentation relative aux systèmes de transport intelligents, en faisant un rapide tour des technologies existantes et des principes de cette tendance qui évolue quotidiennement. Ensuite, nous nous sommes recentrés sur les réseaux de capteurs sans fil, et nous avons étudié l'adéquation de ces derniers avec les STI sur une infrastructure fixe. Nous avons ainsi montré que ces équipements particuliers possèdent l'avantage d'être petits, à bas prix, et d'une logique naturelle distribuée. Enfin, nous avons présenté quelques outils permettant de simuler le trafic routier urbain, et voir ce qu'il était possible ou non d'effectuer.

Qu'il s'agisse des STI en général, ou plus particulièrement de leur utilisation avec des réseaux de capteurs sans fil, la littérature apparait comme étant très dispersée. D'une part, très peu de modèles font référence aux systèmes de contrôle du trafic actuels, encore moins aux nomenclatures en vigueur. D'autre part, il est constaté pour un problème donné que les auteurs proposent rarement une solution complète, à savoir détaillant tout les aspects théoriques ou de mise en place essentiels. Exemple avec le cas des capteurs électromagnétiques, où les auteurs utilisent la technologie mais ne s'occupent pas de détailler des mécanismes pourtant essentiels : la sécurité, le routage, l'économie d'énergie ou encore l'étude de la distance de communication souvent constatée irréaliste.

Les perspectives que laissent entrevoir les réseaux de capteurs sans fil sont nombreuses. D'une part, ces petits équipements n'englobent pas totalement les STI : il serait intéressant d'étudier un réseau distribué plus large, hiérarchisé et formel que les modèles étudiés, afin de pallier aux problèmes rencontrés. D'autre part, l'interaction possible avec des réseaux véhiculaires décuple les possibilités déjà existantes : il serait

intéressant d'étudier les limites et nombreux aspects de ce modèle particulier (qui n'était pas le sujet de ce document).

## Références

- [1] <http://www.cete-normandie-centre.developpement-durable.gouv.fr/les-centres-d-ingenierie-et-de-a511.html>.
- [2] <http://www.paris.fr/pratique/voirie-chantiers-en-cours/boulevard-peripherique/p367>.
- [3] <http://recits.inrets.fr/article32.html>.
- [4] <http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/puehfpp.html>.
- [5] <http://www.auto-bleue.org>.
- [6] <http://www.mobility.ch>.
- [7] <http://senseable.mit.edu/copenhagenwheel/>.
- [8] <http://ntoctsl.groupsite.com/page/fhwa>.
- [9] [http://www.scoot-utc.com/documents/1\\_SCOOT-UTC.pdf](http://www.scoot-utc.com/documents/1_SCOOT-UTC.pdf).
- [10] <http://sourceforge.net/projects/stoplicht/>.
- [11] <http://www.ai.rug.nl/mwiering/>.
- [12] <http://sourceforge.net/projects/morevts/>.
- [13] <http://sumo.sourceforge.net/>.
- [14] <http://www.matsim.org/>.
- [15] <http://www.opendrive.org/download.htm>.
- [16] <http://www.openstreetmap.org/>.
- [17] <http://josm.openstreetmap.de>.
- [18] <http://eworld.sourceforge.net/>.
- [19] Avatarf. <http://sourceforge.net/projects/avartaf/>.
- [20] Cave. <http://sourceforge.net/projects/cautove/>.
- [21] Les transports intelligents. <http://www.transport-intelligent.net/>.
- [22] Performance evaluation of los angeles adaptive traffic control system (atcs) on an arterial corridor. <http://leonard.csusb.edu/outreach/documents/PERFORMANCEEVALUATIONOFLOSANGELES.pdf>.
- [23] Poisson inter-arrival time generation & cdf function. <http://engineering.dartmouth.edu/eric/engs027/outlines/19DiscrEvSim.pdf>.
- [24] La nouvelle dynamique stationnement. *Ville Rail et Transports*, 507 :30–33, Nov. 2010.
- [25] M. Abbas, H. Charara, N. Chaudhary, and Y. s. Jung. *Distributed architecture and algorithm for robust real-time progress evaluation and improvement*. Texas Transportation Institute, Texas A & M University System, 2006.
- [26] Z. R. Abdy. Fuzzy logic traffic signal control.
- [27] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 38(4) :393–422, 2002.
- [28] F. Al-Nasser and H. Rowaihy. Simulation of dynamic traffic control system based on wireless sensor network. In *2011 IEEE Symposium on Computers Informatics (ISCI)*, pages 40–45, Mar. 2011.
- [29] O. Baniyas, R.-E. Precup, and D.-I. Curiae. Problem setting and modeling in vehicles and pedestrians traffic control using sensor networks. In *4th International Symposium on Applied Computational Intelligence and Informatics (SACI 2007)*, pages 83–88, May 2007.
- [30] W. Barnes, T. King, H. Refai, and J. Fagan. A wireless sensor network simulation for highway intersection collision prevention. In *Conference on Intelligent Transportation Systems (ITSC 2007)*, pages 173–177, 2007.
- [31] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo - simulation of urban mobility : An overview. In *The Third International Conference on Advances in System Simulation (SIMUL 2011)*, pages 63–68, Barcelona, Spain, Oct. 2011.
- [32] E. Bingham. Reinforcement learning in neurofuzzy traffic signal control. *European Journal of Operational Research*, 131(2) :232–241, 2001.
- [33] F. Boillot, J. Blosserville, J. Lesort, V. Motyka, M. Papageorgiou, and S. Sella. Optimal signal control of urban traffic networks. Number 472, pages 182–186, 1992.
- [34] J. Cao, H. Wu, X. Liu, and Y. Lai. isensnet : an infrastructure for research and development in wireless sensor networks. *Frontiers of Computer Science in China*, 4(3) :339–353, 2010.
- [35] Chandra, Reggie and Gregory, Chris. Insync adaptive traffic signal technology : Real-time artificial intelligence delivering real-world results. July 2010.
- [36] E. C.-P. Chang. Guidelines for actuated controllers in coordinated systems. *Transportation Research Record : Journal of the Transportation Research Board*, (1554) :61–73, 1996.
- [37] G.-I. Chang, M. Vasudevan, and C.-c. Su. Modelling and evaluation of adaptive bus-preemption control with and without automatic vehicle location systems. *Transportation Research Part A : Policy and Practice*, 30(4) :251–268, July 1996.
- [38] B. M. Chard and C. J. Lines. Transyt - the latest developments. *Traffic engineering & control*, 28(7-8) :387–390, 1987.
- [39] W. Chen, L. Chen, Z. Chen, and S. Tu. Wits : A wireless sensor network for intelligent transportation system. In *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2006)*, volume 2, pages 635–641, June 2006.
- [40] X.-F. Chen and Z.-K. Shi. Real-coded genetic algorithm for signal timing optimization of a single intersection. In *2002 International Conference on Machine Learning and Cybernetics*, volume 3, pages 1245–1248, 2002.
- [41] S. Cheung, S. Coleri, B. Dundar, S. Ganesh, C. Tan, and P. Varaiya. Traffic measurement and vehicle classification with single magnetic sensor. *Transportation Research Record : Journal of the Transportation Research Board*, 1917(-1) :173–181, 2005.
- [42] S. Coleri, S. Y. Cheung, and P. Varaiya. Sensor networks for monitoring traffic. In *In Allerton Conference on Communication, Control and Computing*, 2004.
- [43] I. Corredor, A. García, J. Martínez, and P. López. Wireless sensor network-based system for measuring and monitoring road traffic. 2008.

- [44] D. W. Dey, S. Fitzsimons, A. Morris, and D. Ng. Adaptive traffic signal interconnect in menlo park and sunnyvale, ca. 2002.
- [45] M. Ducarne and T. Périneau. Commande adaptative et optimale d'un carrefour à feu. Technical report, 2007. [http://ww2.eivp-paris.fr/chachoua/TER2007/%5BN%B026%5D\\_RapportTER\\_Ducarne\\_P%E9rineau.pdf](http://ww2.eivp-paris.fr/chachoua/TER2007/%5BN%B026%5D_RapportTER_Ducarne_P%E9rineau.pdf).
- [46] C. K. G., L. J. Z., and W. C. E. Development of guidelines for implementing computerized timing designs at traffic actuated signals, two volumes on arterial system implementation, transp. res. center, gainesville. 1989.
- [47] N. Gartner, F. Pooran, and C. Andrews. Implementation of the opac adaptive control strategy in a traffic signal network. In *IEEE Intelligent Transportation Systems*, pages 195–200, 2001.
- [48] R. Gordon, W. Tighe, U. S. F. H. A. O. of Operations, D. E. Associates, and I. Siemens. *Traffic control systems handbook*. US Dept. of Transportation, Federal Highway Administration, Office of Operations, 2005. <http://ops.fhwa.dot.gov/publications/fhwahop06006/>.
- [49] K. L. Head, P. B. Mirchandani, and D. Sheppard. Hierarchical framework for real-time traffic control. *Transportation Research Record 1360, Transportation Research Board, National Research Council, Washington DC*, 1992.
- [50] J. Henry, J. Farges, and J. Tuffal. The prodyn real time traffic algorithm. *4th Conference on Control Transportation System*, (472) :305–309, 1983.
- [51] W. Homburger, J. Kell, and P. D.D. Fundamentals of traffic engineering, institute of transportation studies, university of california at berkeley, (13th edition). 1992.
- [52] D. Houli, L. Zhiheng, and Z. Yi. Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network. *EURASIP J. Adv. Signal Process*, 2010 :7 :1–7 :7, Mar. 2010.
- [53] N. Hounsell, B. Shrestha, F. McLeod, S. Palmer, T. Bowen, and J. Head. Using global positioning system for bus priority in london : traffic signals close to bus stops. *Intelligent Transport Systems, IET*, 1(2) :131–137, June 2007.
- [54] N. B. Hounsell, F. N. McLeod, and B. P. Shrestha. Bus priority at traffic signals : Investigating the options. In *IEE Conference Publication*, number 501, pages 287–294. Institution of Engineering and Technology, 2004.
- [55] IBM. Sondage mondial ibm sur la pénibilité du navettage : congestion routière en baisse, pénibilité nettement en hausse, June 2011. <http://www.ibm.com/news/ca/fr/2011/09/08/u251344d61892u91.html>.
- [56] J. H. Kell and I. J. Fullerton. Manual of traffic signal design, 3rd ed. englewood cliffs, nj : Inst. transp. eng., prentice-hall. 1998.
- [57] A. Kesharwani, V. Sadaphal, and M. Natu. Empowering bus transportation system using wireless sensor networks. In *7th International Conference on High Performance Computing (HiPC 2010)*, Goa, India, Dec. 2010.
- [58] A. Knaian. *A wireless sensor network for smart roadbeds and intelligent transportation systems*. PhD thesis, Citeseer, 2000.
- [59] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Rössel, W. Tuchscheerer, P. Wagner, and R. Wöslér. Simulation of modern traffic lights control systems using the open source traffic simulation sumo. In *3rd Industrial Simulation Conf*, 2005.
- [60] O. F. L. The traffic signal book. englewood cliffs, nj : Prentice- hall. 1993.
- [61] C. Lee. Fuzzy logic in control systems : fuzzy logic controller. ii. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2) :419–435, mar/apr 1990.
- [62] M. Lescieux. Application à la commande floue. [http://auto.polytech.univ-tours.fr/automatique/AUA/ressources/Introduction\\_logique\\_floue.ppt](http://auto.polytech.univ-tours.fr/automatique/AUA/ressources/Introduction_logique_floue.ppt).
- [63] Z. Liao and L. Zhao. Wireless sensor networks help to improve the traffic safety in residential communities. In *6th International Conference on ITS Telecommunications Proceedings*, pages 973–978, June 2006.
- [64] Z. Liao and L. Zhao. Wireless sensor networks help to improve the traffic safety in residential communities. In *6th International Conference on ITS Telecommunications*, pages 973–978, 2006.
- [65] F. B. Lin. Optimal timing settings and detector lengths of presence model full-actuated control. *Transportation Research Record*, 1010 :37–45, 1985.
- [66] B. Liu and W. Liu. Evaluation of traffic control methods at traffic circles. In *Control and Decision Conference (CCDC), 2011 Chinese*, pages 3371–3377, May 2011.
- [67] W. Ma and X. Yang. Design and evaluation of an adaptive bus signal priority system base on wireless sensor network. In *11th International IEEE Conference on Intelligent Transportation System (ITSC 2008)*, pages 1073–1077, Oct. 2008.
- [68] R. R. McShane W.R. Traffic engineering, prentice hall. 1990.
- [69] L. E. Y. Mimbela and L. A. Klein. *Summary of vehicle detection and surveillance technologies used in intelligent transportation systems*. Federal Highway Administration, Intelligent Transportation Systems Joint Program Office, 2007.
- [70] Minnesota Department of Transportation. Traffic signals 101, 2006. <http://www.dot.state.mn.us/trafficeng/publ/signals101/index.html>.
- [71] National Electrical Manufacturers Association. Nema ts2 traffic controller assemblies with ntcip requirements, 1998.
- [72] OMT. Conférence sur les déplacements écologiques en europe, 2006. <http://www.busandcoach.travel/download/factsheets/frgreen.pdf>.
- [73] D. R. Pritchard and R. N. Buliung. Transportation analysis and research using free/open source software. 2007.
- [74] D. Robertson and R. Bretherton. Optimizing networks of traffic signals in real time-the scoot method. *IEEE Transactions on Vehicular Technology*, 40(1) :11–15, Feb. 1991.
- [75] H. Sawant, J. Tan, Q. Yang, and Q. Wang. Using bluetooth and sensor networks for intelligent transportation systems. In *7th International IEEE Conference on Intelligent Transportation Systems*, pages 767–772, Oct. 2004.
- [76] M. Selinger and L. Schmidt. A review of the cost, maintenance and reliability of popular adaptive traffic control technologies. *Adaptive Traffic Control Systems in the United States*, sep 2009.
- [77] P. Shan-Chen, S. Yan-Sen, L. Ye, S. Jie, and Z. Huai-Zhou. A study on the model of the traffic signs in the traffic circle. In *International Conference on Information Engineering and Computer Science (ICIECS 2009)*, pages 1–4, Dec. 2009.
- [78] A. Sims and K. Dobinson. The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on Vehicular Technology*, 29(2) :130–137, May 1980.
- [79] A. Skabardonis, R. L. Bertini, and B. R. Gallagher. Development and application of control strategies for signalized intersections in coordinated systems. *Transportation research record*, (1634) :110–117, 1998.
- [80] J. Spall and D. Chin. Traffic-responsive signal timing for system-wide traffic control. *Transportation Research Part C : Emerging Technologies*, 5(3-4) :153–163, 1997.
- [81] K. J. T. and C. K. G. Evaluation and design of maximum green time settings for traffic actuated control. *Transportation Research Record*, 1852 :246–255, 2003.

- [82] S. Takahashi, H. Nakamura, H. Kazama, and T. Fujikura. Genetic algorithm approach for adaptive offset optimization for the fluctuation of traffic flow. In *The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 768 – 772, 2002.
- [83] F. Tang, M. Li, C. Weng, C. Zhang, W. Zhang, H. Huang, and Y. Wang. Combining wireless sensor network with grid for intelligent city traffic. In *11th Asia-Pacific Conference, ACSAC 2006*, volume 4186, pages 260–269, Shanghai, China, Sept. 2006.
- [84] M. Tubaishat, Q. Qi, Y. Shang, and H. Shi. Wireless sensor-based traffic light control. In *5th IEEE Conference on Consumer Communications and Networking (CCNC 2008)*, pages 702–706, 2008.
- [85] M. Tubaishat, Y. Shang, and H. Shi. Adaptive traffic light control with wireless sensor networks. In *4th IEEE Conference on Consumer Communications and Networking*, 2007.
- [86] University of Idaho-Moscow. Traffic signal training. <http://www.webs1.uidaho.edu/niattproject/>.
- [87] W. Wei and Y. Zhang. Fl-fn based traffic signal control. In *IEEE International Conference on Fuzzy Systems*, volume 1, pages 296 –300, 2002.
- [88] Y. Wen, J. Pan, and J. Le. Survey on application of wireless sensor networks traffic monitoring. In *First International Conference on Transportation*, volume 246, pages 2079–2084. American Society of Civil Engineers, 1801 Alexander Bell Drive Reston VA 20191-4400 USA,, 2007.
- [89] C. Wenjie, C. Lifeng, C. Zhanglong, and T. Shiliang. A realtime dynamic traffic control system based on wireless sensor network. In *International Conference Workshops on Parallel Processing (ICPP 2005)*, pages 258 – 264, June 2005.
- [90] C. Wenjie, G. Liqiang, C. Zhilei, C. Zhanglong, and T. Shiliang. An intelligent guiding and controlling system for transportation network based on wireless sensor network technology. In *Fifth International Conference on Computer and Information Technology (CIT 2005)*, pages 810 – 814, Sept. 2005.
- [91] M. Wiering. Multi-agent reinforcement learning for traffic light control. In *Seventeenth International Conference on Machine Learning*, pages 1151–1158, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [92] M. Wiering, V. J. V., V. J., and A. Koopman. Intelligent traffic light control. In *Technical report UU-CS-2004-029.*, 2004.
- [93] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman. Simulation and optimization of traffic in a city. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 453 – 458, June 2004.
- [94] L. Xiao, X. Peng, Z. Wang, B. Xu, and P. Hong. Research on traffic monitoring network and its traffic flow forecast and congestion control model based on wireless sensor networks. *Measuring Technology and Mechatronics Automation, International Conference on*, 1 :142–147, 2009.
- [95] Y. L. Y., Z. Y. L., and Q. L. C. Research and implementation of transportation monitoring system based on distributed wireless sensor network. *Computer Engineering*, (32) :249–251, 2006.
- [96] K. Yousef, J. Al-Karaki, and A. Shatnawi. Intelligent traffic light flow control system using wireless sensors networks. *Journal of Information Science and Engineering*, 26(3) :753–768, 2010.
- [97] X. Zeng and H. Zheng. The intelligent control and modeling of a traffic circle. In *International Conference on Information Engineering and Computer Science (ICIECS 2009)*, pages 1 –4, Dec. 2009.
- [98] G. Zhang and W. Y. Optimizing minimum and maximum green time settings for traffic actuated control at isolated intersections. *Intelligent Transportation Systems, IEEE Transactions on*, (99) :1–10, 2011.
- [99] Q. Zheng and M. Li. The methods of traffic circle problem. In *International Conference on Logistics Engineering and Intelligent Transportation Systems (LEITS 2010)*, pages 1 –4, Nov. 2010.
- [100] B. Zhou, J. Cao, and H. Wu. Adaptive traffic light control of multiple intersections in wsn-based its. In *Vehicular Technology Conference (VTC Spring)*, pages 1–5.
- [101] B. Zhou, J. Cao, X. Zeng, and H. Wu. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In *Vehicular Technology Conference Fall (VTC 2010-Fall)*, pages 1–5.
- [102] F. Zou, B. Yang, and Y. Cao. Traffic light control for a single intersection based on wireless sensor network. In *9th International Conference on Electronic Measurement & Instruments (ICEMI 2009)*, pages 1–1040, 2009.



## A SUMO / NETGEN : Création d'une simulation

Ci-dessous les fichiers XML nécessaires à la création d'une intersection conforme au modèle représenté à la figure 8.

### projet.nod.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<nodes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/nodes_file.xsd">
  <node id="0" x="0.0" y="0.0" type="traffic_light"/>

  <node id="1" x="-500.0" y="0.0" type="priority"/>
  <node id="2" x="+500.0" y="0.0" type="priority"/>
  <node id="3" x="0.0" y="-500.0" type="priority"/>
  <node id="4" x="0.0" y="+500.0" type="priority"/>

  <node id="51" x="-510.0" y="0.0" type="priority"/>
  <node id="52" x="+510.0" y="0.0" type="priority"/>
  <node id="53" x="0.0" y="-510.0" type="priority"/>
  <node id="54" x="0.0" y="+510.0" type="priority"/>
</nodes>
```

### projet.edg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<edges xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/edges_file.xsd">
  <edge id="1i" from="1" to="0" priority="78" numLanes="4" speed="19.444" />
  <edge id="1o" from="0" to="1" priority="46" numLanes="4" speed="11.111" />

  <edge id="2i" from="2" to="0" priority="78" numLanes="4" speed="19.444" />
  <edge id="2o" from="0" to="2" priority="46" numLanes="4" speed="11.111" />

  <edge id="3i" from="3" to="0" priority="78" numLanes="4" speed="19.444" />
  <edge id="3o" from="0" to="3" priority="46" numLanes="4" speed="11.111" />

  <edge id="4i" from="4" to="0" priority="78" numLanes="4" speed="19.444" />
  <edge id="4o" from="0" to="4" priority="46" numLanes="4" speed="11.111" />

  <edge id="51i" from="1" to="51" priority="78" numLanes="3" speed="19.444" />
  <edge id="51o" from="51" to="1" priority="46" numLanes="3" speed="11.111" />

  <edge id="52i" from="2" to="52" priority="78" numLanes="3" speed="19.444" />
  <edge id="52o" from="52" to="2" priority="46" numLanes="3" speed="11.111" />

  <edge id="53i" from="3" to="53" priority="78" numLanes="3" speed="19.444" />
  <edge id="53o" from="53" to="3" priority="46" numLanes="3" speed="11.111" />

  <edge id="54i" from="4" to="54" priority="78" numLanes="3" speed="19.444" />
  <edge id="54o" from="54" to="4" priority="46" numLanes="3" speed="11.111" />
</edges>
```

### projet.con.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<connections xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/connections_file.xsd">
  <connection from="1i" to="4o" fromLane="3" toLane="2"/>
  <connection from="2i" to="3o" fromLane="3" toLane="2"/>
  <connection from="3i" to="1o" fromLane="3" toLane="2"/>
  <connection from="4i" to="2o" fromLane="3" toLane="2"/>

  <connection from="1i" to="2o" fromLane="1" toLane="0"/>
  <connection from="2i" to="1o" fromLane="1" toLane="0"/>
  <connection from="3i" to="4o" fromLane="1" toLane="0"/>
  <connection from="4i" to="3o" fromLane="1" toLane="0"/>
  <connection from="1i" to="2o" fromLane="2" toLane="1"/>
  <connection from="2i" to="1o" fromLane="2" toLane="1"/>
  <connection from="3i" to="4o" fromLane="2" toLane="1"/>
  <connection from="4i" to="3o" fromLane="2" toLane="1"/>

  <connection from="1i" to="3o" fromLane="1" toLane="0"/>
  <connection from="2i" to="4o" fromLane="1" toLane="0"/>
  <connection from="3i" to="2o" fromLane="1" toLane="0"/>
  <connection from="4i" to="1o" fromLane="1" toLane="0"/>
</connections>
```

### projet.netc.xml

```

<?xml version="1.0" encoding="iso-8859-1"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/netconvertConfiguration.xsd">
  <input>
    <node-files value="cross.nod.xml"/>
    <edge-files value="cross.edg.xml"/>
    <connection-files value="cross.con.xml"/>
  </input>
  <output>
    <output-file value="cross.net.xml"/>
  </output>
</configuration>

```

La commande suivante permet de convertir cet ensemble de fichier en un fichier **projet.net.xml** :

```
netconvert -c projet.netc.xml
```

Enfin, le fichier **projet.rou.xml** met en place la demande :

```

<routes>
  <vType id="car" accel="0.8" decel="4.5" sigma="0.5" length="5"
    minGap="2.5" maxSpeed="16.67" guiShape="passenger"/>

  <route id="right" edges="51o 1i 2o 52i" />
  <route id="left" edges="52o 2i 1o 51i" />
  ...

  <vehicle id="0" type="car" route="right" depart="0" />
  <vehicle id="1" type="car" route="left" depart="0" />
  <vehicle id="2" type="car" route="right" depart="1" />
  ...
</routes>

```

Au final, le fichier de configuration de la simulation **projet.sumo.cfg** n'a plus qu'à être défini :

```

<?xml version="1.0" encoding="iso-8859-1"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">
  <input>
    <net-file value="cross.net.xml"/>
    <route-files value="cross.rou.xml"/>
    <additional-files value="cross.det.xml"/>
    <gui-settings-file value="gui-settings.cfg"/>
  </input>
  <output>
    <tripinfo-output value="output-tripinfos.xml"/>
  </output>
  <time>
    <begin value="0"/>
  </time>
  <report>
    <no-step-log value="true"/>
  </report>
</configuration>

```

Précisons qu'ici les détecteurs sont mis en place conformément à la figure 8, avec une position fixée.

```

<sumo-detectors>
  <e1Detector id="0" lane="4i_1" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="1" lane="2i_1" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="2" lane="1i_1" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="3" lane="3i_1" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="4" lane="4i_2" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="5" lane="2i_2" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="6" lane="1i_2" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="7" lane="3i_2" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="8" lane="4i_3" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="9" lane="2i_3" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="10" lane="1i_3" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="11" lane="3i_3" pos="467" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="12" lane="4i_3" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="13" lane="4i_2" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="14" lane="4i_1" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="15" lane="3i_3" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="16" lane="3i_2" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="17" lane="3i_1" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="18" lane="2i_3" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="19" lane="2i_2" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="20" lane="2i_1" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="21" lane="1i_3" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="22" lane="1i_2" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
  <e1Detector id="23" lane="1i_1" pos="500" freq="1" file="cross.out" friendlyPos="x"/>
</sumo-detectors>

```



