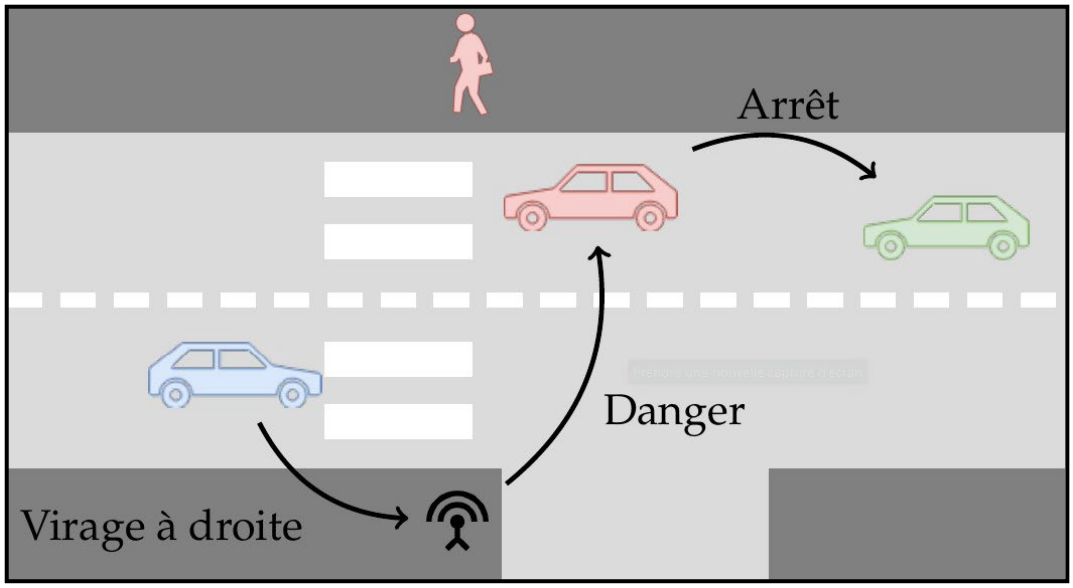
Prédiction de comportement des algorithmes coopératifs dans les réseaux véhiculaires

Guillaume Béduneau

Par **Guillaume BÉDUNEAU**

Prédiction de comportement des algorithmes coopératifs dans les réseaux véhiculaires

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 31 mars 2023
Spécialité : Informatique : Unité de recherche Heudyasic (UMR-7253)
D2760

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

Présentée par Monsieur

Guillaume BÉDUNEAU

En vue d'obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

Spécialité : Informatique

École doctorale : Sciences pour l'ingénieur

Unité de recherche : Laboratoire Heudiasyc

Équipe de recherche : SCOP

Prédiction de comportement des algorithmes coopératifs dans les réseaux véhiculaires

Thèse soutenue à Compiègne

Le 31 mars 2023

Devant le jury composé de :

Pr. Hervé RIVANO, INSA Lyon	(Rapporteur)
Pr. Sidi-Mohamed SENOUCI, Université de Bourgogne	(Rapporteur)
Dr. Reine TALJ, Chargée de recherche CNRS, HdR	(Examinatrice)
Pr. Véronique VÈQUE, Université Paris Saclay	(Examinatrice)
Pr. Bertrand DUCOURTHIAL, Université de Technologie de Compiègne Dr. Ghada JABER, Université de Technologie de Compiègne	(Directeur de thèse) (Directrice de thèse)

Remerciements

Le travail mené dans cette thèse a été réalisé au sein du laboratoire Heudiasyc de l'Université de Technologie de Compiègne (UTC). Il a bénéficié de l'aide et du support de nombreuses personnes aussi bien au quotidien que lors d'événements ponctuels.

Je tiens tout d'abord à remercier Bertrand Ducourthial, mon directeur de thèse depuis le démarrage de cette aventure, pour son accompagnement et le temps passé lors de toutes les étapes de la thèse. Il a fait preuve d'une grande compréhension lors des moments difficiles et s'est montré attentionné au bien-être de toute l'équipe impliquée dans cette thèse. Je souhaite également remercier Ghada Jaber, qui, malgré son arrivée plus tardive dans la direction de la thèse a su s'intégrer au travail déjà réalisé et a apporté à la fois un recul appréciable, et une expertise complémentaire dans d'autres domaines réseau. La totalité de la thèse a bénéficié d'une direction de thèse efficace, y compris dans les moments difficiles. Ils ont constamment su guider les recherches et partager leur connaissance du monde universitaire tout en poussant la qualité de ces travaux de thèse à son maximum.

Mes remerciements vont également aux rapporteurs, Sidi-Mohamed SENOUCI et Hervé Rivano, parce qu'ils ont accepté de relire mon travail, et pour leur participation à son amélioration tant avant la soutenance que pendant, par leurs questions et remarques. Je tiens également à remercier Véronique Vèque, qui a présidé la soutenance de ma thèse, s'est montrée à la fois particulièrement intéressée et constructive, m'a aidé à me sentir à l'aise au cours de toute la soutenance et a animé les questions du jury. Enfin, je remercie également Reine Talj, examinatrice du jury de thèse, pour ses questions et le partage de son expertise à la fin de la soutenance.

Les travaux réalisés durant cette thèse ont bénéficié de l'aide de tout le personnel support du laboratoire Heudiasyc, notamment du point de vue administratif, avec le concours notable de Nathalie Alexandre et Sabine Collé, et du point de vue technique, avec l'aide de Thierry Monglon, Gildas Bayard, Stéphane Bonnet, qui m'ont facilité l'usage des plateformes technologiques, à la fois pour les expériences réelles et les expériences virtuelles. Au sein du laboratoire Heudiasyc, je remercie également tous les doctorants, stagiaires et permanents qui ont su participer à créer une ambiance de travail à la fois efficace et agréable.

Le soutien de mes proches a également été crucial, notamment pour surmonter les difficultés personnelles qui sont survenues durant ma thèse, avec notamment la force et le recul donné par ma femme Océane, notre fils Lucas, né en 2022, qui, à la fois par et malgré sa maladie, nous a donné le courage d'avancer malgré l'adversité, et toute notre famille, qui s'est assurée au maximum de notre bien-être, sans lequel nos thèses n'auraient probablement pas pu aboutir en 2023.

Enfin, les moyens financiers déployés sur ces travaux ont été fournis par l'Agence Innovation Défense (AID) et le labex Maîtrise des Systèmes de Systèmes Technologiques (MS2T), qui m'ont offert des conditions de travail tout à fait confortables auxquelles tous les doctorants contractuels n'ont pas le droit.

Résumé

L'arrivée de technologies de plus en plus avancées dans les véhicules du quotidien (voitures, camions, bus) permet, depuis plusieurs décennies, d'améliorer le confort des usagers de la route et leur sécurité. Ces technologies permettent également d'échanger des informations entre véhicules (communications Véhicule à Véhicule V2V) ou avec l'infrastructure routière (communications Véhicule à Infrastructure V2I). Des applications coopératives destinées à améliorer la sécurité routière ou à optimiser l'utilisation du réseau routier peuvent alors se baser sur de tels échanges.

Pour la sécurité de leurs utilisateurs, on attend de ces applications qu'elles soient réactives et aient un comportement prédictible. En effet, les conséquences d'un comportement erratique d'une application coopérative de sécurité peuvent être désastreuses, tandis que son utilité est limitée si sa trop faible réactivité ne permet pas d'anticiper suffisamment les risques.

Dans ce contexte, cette thèse cherche comment assurer la validation des applications de coopération entre véhicules au sein d'un réseau routier déjà complexe. En effet, la validation des applications véhiculaires coopératives pose différents problèmes. Elle peut être théorique, par exemple à travers une preuve de l'algorithme, mais cette approche s'avère parfois manquer de réalisme. Cela la rend alors peu représentative du comportement de l'application au cours d'un déploiement réel. La validation empirique d'une application est possible mais ces expériences sont souvent difficiles à extrapoler à un déploiement réel. Leur coût et les difficultés logistiques qu'elles imposent sont également parfois dissuasifs.

Les difficultés de validation d'applications véhiculaires coopératives nous conduisent ensuite à proposer une méthode de prédiction de performance conçue pour simplifier une telle validation. Cette méthode est basée sur une observation réelle de réseau véhiculaire et sur une analyse théorique du fonctionnement de l'algorithme de coopération. Une évaluation empirique de la méthode de prédiction proposée est également menée, afin d'en vérifier l'efficacité. Elle est finalement appliquée dans un cas concret, démontrant son fonctionnement face à un réel problème de sécurité routière identifié par la communauté scientifique qui concerne l'amélioration de la sécurité des piétons sur la route.

Abstract

The arrival of increasingly advanced technologies in everyday vehicles (cars, trucks, buses) has made it possible, for several decades, to improve the comfort of road users and their safety. These technologies also enable to exchange information between vehicles (V2V Vehicle-to-Vehicle communications) or with the road infrastructure (V2I Vehicle-to-Infrastructure communications). Cooperative applications intended to improve road safety or to optimize the use of the road network can then be based on such exchanges.

For the safety of their users, these applications are expected to be responsive and have predictable behavior. Indeed, the consequences of an erratic behavior of a cooperative security application can be disastrous, while its usefulness is limited if it is not reactive enough to enable to anticipate the risks.

In this context, this thesis seeks how to ensure the validation of cooperative applications between vehicles within an already complex road network. Indeed, the validation of cooperative vehicular applications poses various problems. It can be theoretical, for example through a proof of the algorithm, but this approach sometimes turns out to be unrealistic. This then makes it unrepresentative of the behavior of the application during an actual deployment. Empirical validation of an application is possible but the involved experiments are often difficult to extrapolate to a real deployment. Their cost and the logistical difficulties they impose may also be dissuasive.

The difficulties of validating cooperative vehicular applications then lead us to propose a performance prediction method designed to simplify such validation. This method is based on a real observation of a vehicular network and on a theoretical analysis of the behavior of the cooperative algorithm. An empirical evaluation of the proposed prediction method is also conducted, in order to verify its effectiveness. It is finally applied in a concrete use case, demonstrating its operation in the face of a real road safety problem identified by the scientific community, which concerns the improvement of pedestrian safety on the road.

Table des matières

Table des abréviations

Abréviation	**Signification**
AID	Agence Innovation Défense
ANR	Agence Nationale de la Recherche
AODV	Ad-hoc On-demand Distance Vector
BSM	Basic Safety Message
CAM	Cooperative Awareness Message
C-ITS	Cooperative ITS
CNRS	Centre National de la Recherche Scientifique
CPM	Collective Perception Message
C-V2X	Cellular V2X
DENM	Decentralized Event Notification Message
DSRC	Dedicated Short-Range Communication
ETSI	European Telecommunications Standards Institute
GPS	Global-Positioning System
Heudiasyc	Heuristique et diagnostic des systèmes complexes
ITS	Intelligent Transportation System
MANET	Mobile Ad-hoc NETwork
MPR	Multi-Point Relay
MS2T	Maîtrise des Systèmes de Systèmes Technologiques
OLSR	Optimized Link State Routing Protocol
PBU	Pedestrian Body Unit
PND	Proactive Neighborhood Discovery
PoC	Proof of Concept
PTD	Proactive Triangle Detection
RSU	Road Side Unit
TTC	Time To Collision
TVG	Time Varying Graphs
UE	Union Européenne
UTC	Université de Technologie de Compiègne
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to everything
VANET	Vehicular Ad-hoc NETwork
VRU	Vulnerable Road User
ZRR	Zone à Régime Restrictif

Table des symboles

δ	Fonction de durée de transfert d'un p-graphe
e	arête d'un graphe
i	entier
id	entier
j	entier
l	entier
M	Matrice
m	message
$\mathcal{F}$	Famille de p-graphes dynamiques
$\mathcal{G}$	Graphe dynamique
$\mathcal{G}^p$	p-graphe dynamique
n	entier
N	entier
p	entier
q	entier
s	entier
t	date
nseq	entier
pld	chaîne de caractères
rel	entier
sel	entier
source	entier
ζ	Fonction de latence d'un TVG
d	Durée
U	Ensemble
V	Ensemble

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

1.1 Contexte général : système de transport et société

1.1.1 Enjeux des nouvelles technologies dans le réseau routier

Dans les pays développés, le réseau routier constitue une infrastructure stratégique cruciale. Il joue ainsi un rôle économique important en permettant le transport des marchandises, mais aussi les trajets domicile-travail de la population. En 2019, 75 % du transport terrestre des marchandises de l'Union Européenne (UE) était effectué par la route [**?**], et plus de 80 % des passagers utilisaient un transport routier en 2012 [**?**].

En plus de son rôle économique, le réseau routier a un rôle social important. En effet, la voiture individuelle est devenue, au cours du XX^{e} siècle véritablement iconique de la réussite sociale et de la liberté. La large gamme de voitures proposées par les constructeurs automobiles permet d'ailleurs d'assurer une personnalisation de l'équipement de chaque véhicule à son propriétaire.

L'arrivée des nouvelles technologies dans le quotidien a aussi été l'occasion d'ajouter des équipements ou d'améliorer les fonctionnalités d'équipements existants. Ces technologies se sont intégrées dans toutes les composantes de la voiture, pour leur apporter une valeur ajoutée aux yeux des clients.

Dans les pays développés, les sociétés civiles sont de plus en plus demandeuses de changements, qui concernent notamment l'industrie. Ainsi, les citoyens se sentent concernées par l'impact environnemental et social de ce qu'ils consomment, et attendent des industriels et des autorités une prise en compte plus systématique de ces enjeux. La production et l'utilisation des véhicules sont donc soumises à ces préoccupations et doivent s'adapter pour les prendre en compte.

Les industries des systèmes de transport routier sont également soumises à des pressions qui leur sont plus spécifiques au sujet de la sécurité routière. En effet, les transports routiers sont responsables, en Europe d'environ 50 morts par million d'habitants dans l'UE [**?**], ce qui représente, pour la France, autant que le cancer du poumon. La forte mortalité des routes au milieu du XX^{e} siècle pousse, depuis cette époque, la société civile et le régulateur à affecter toujours plus d'efforts pour la sécurité des véhicules et des usagers de la route. Ces efforts se sont montrés, au départ très efficaces (le nombre de morts sur les routes de France passe de plus de 18 000 en 1972 à moins de 9 000 en 2000 malgré un triplement du trafic routier). Ils commencent cependant à montrer leurs limites depuis une dizaine d'années en France, avec une baisse très limitée de la mortalité qui stagne autour de 3 000 morts depuis 2010 [**?**].

En plus des systèmes de sécurité routière, un système juridique complexe de responsabilités et de présomption a ainsi été mis en place pour la gestion des accidents de la route. Il est basé sur la responsabilité civile des conducteurs et propriétaires de véhicules. Ainsi, chaque propriétaire de véhicule a l'obligation de disposer d'une assurance couvrant tous les dommages éventuellement produits par son utilisation. Les conducteurs de véhicules sont automatiquement présumés responsables des accidents lorsqu'ils n'impliquent pas d'autres véhicules, et les responsabilités des accidents sans dommages corporels sont gérées directement à l'amiable par les assurances via un mécanisme de constat unifié à travers l'UE (constat européen) introduit en 2006.

Face à l'apparition ces dernières décennies de nouvelles technologies et de nouvelles préoccupations à la fois sociales et environnementales, il semble nécessaire, pour le système de transport des pays industrialisés, de se moderniser. Cela passe notamment par l'intégration de nouveaux équipements capables de répondre aux inquiétudes de leur population. Cette modernisation doit prendre en compte l'aspect stratégique de l'infrastructure routière et de son utilisation, et, par conséquent, les risques inhérents à chaque changement qui la concerne.

1.1.2 Coopération véhiculaire

La modernisation des systèmes de transport routier est une dynamique qui est en réalité déjà amorcée. En effet, l'apparition des navigateurs GPS (Global-Positioning System) qui suit l'ouverture des signaux aux civils en 1983 se matérialise dès 1990 par l'intégration dans la Mazda Eunos Cosmo. Le développement des réseaux de téléphones portables a également eu un impact fort sur le réseau routier. De nouvelles législations ont été nécessaires, interdisant par exemple l'usage du téléphone portable pendant la conduite en 2004 en France.

Sur le réseau routier, la circulation est depuis ses débuts basée sur la coopération entre les conducteurs des véhicules, qui communiquent à travers des gestes (gestion de certaines intersections par la courtoisie), des signaux lumineux (clignotants, feux de stop, appels de phares) et sonores (avertisseur sonore de type klaxon). Cette coopération permet à la fois une meilleure prise en compte des dangers (freinage brusque, manœuvre d'évitement par exemple) et de gérer les situations complexes (intersection très chargée ou déformation de chaussée, par exemple). Cependant, ces capacités de communication sont assez limitées, et il est parfois déjà difficile pour les conducteurs de les interpréter correctement : il n'y a pas de différences de communication (uniquement le clignotant gauche dans les deux cas) entre un véhicule indiquant qu'il ralentit avec son frein moteur pour tourner à gauche et un véhicule indiquant qu'il accélère pour effectuer un dépassement. L'utilisation de systèmes plus modernes (transmission via un protocole de communication sans fil) peut assurer le transit, en un temps réduit, d'une information plus riche (indiquant la vitesse cible d'un freinage, l'intention réelle derrière l'allumage d'un clignotant) d'un véhicule à un autre.

L'environnement routier fait l'objet d'une très intense réglementation, avec par exemple, en France, un code législatif distinct (le code de la route) pour exprimer les règles de circulation uniquement. Ces réglementations assurent la pérennité et l'efficacité du réseau réseau routier en tant qu'infrastructure critique, mais peuvent constituer un obstacle à l'innovation dans le domaine des transports. Ainsi, l'ajout d'un nouvel équipement de série dans un véhicule par son constructeur nécessite généralement, en France, une nouvelle demande d'homologation auprès du Centre National de Réception des Vé-

hicules (CNRV). Cette homologation se fera en partie sur la base de tests effectués sur le véhicule, réalisés en laboratoires par l'UTAC (Union Technique de l'Automobile, du motocycle et du Cycle), sur un autodrome situé à Montlhéry, en Ile de France.

1.1.3 Développement d'applications véhiculaires coopératives

Dans le contexte juridique du réseau routier, il est en souvent plus simple, pour le conducteur, d'ajouter temporairement un équipement réalisant la fonctionnalité dont il a besoin que pour un constructeur automobile d'ajouter de façon permanente des équipements assurant cette fonctionnalité dans une nouvelle version de véhicule. Ainsi, la démocratisation du smartphone a permis l'apparition d'applications destinées à être utilisées en voiture. Ces applications peuvent utiliser la connexion à Internet du smartphone (données mobiles) pour transmettre et récupérer des informations d'intérêt en plus des informations de navigation (itinéraire). C'est par exemple l'approche de l'application mobile Waze [**?**], qui permet aux usagers des routes de s'échanger des informations sur les limitations de vitesse d'une route, la fermeture de voies d'une autoroute ou la présence de bouchons. Cependant, le fonctionnement de Waze pose de nombreux problèmes, liés au respect de la vie privée [**?**] des usagers du réseau ou à l'interface utilisateur [**?**].

En choisissant une approche plus intégrée au véhicule que celle d'une application pour smartphone, on peut en améliorer l'efficacité. Cela peut être réalisé en utilisant directement des communications sans fil d'un véhicule à l'autre, au sein d'un réseau ad hoc (sans infrastructure prédéfinie) véhiculaire. Ce type de réseau est également appelé VANET (Vehicular Ad-hoc NETwork). Il est ainsi possible de connecter directement les capteurs du véhicule, ses calculateurs et le système de communication, ce qui peut permettre un partage automatisé (sans humain dans la boucle) de certaines informations. Les données partagées peuvent cependant être paramétrables par le conducteur ou le propriétaire du véhicule. Cela réduit les risques de faire perdre de l'attention au conducteur, et permet d'accélérer le transfert en limitant les réactions humaines souvent plus lentes que les ordinateurs dans ce contexte. Cette approche permet également de fiabiliser les connaissances en comparant les observations des capteurs du véhicule avec celles menées sur les autres véhicules. Par ailleurs, il devient alors possible de communiquer directement d'un véhicule à l'autre, par exemple lorsqu'aucune connexion à Internet n'est disponible (tunnels, zones blanches, mauvaises conditions météo). Le matériel nécessaire à cette fin est plutôt limité et peu coûteux, il n'a pas besoin de grandes capacités de calcul ni de capteurs supplémentaires, et ne nécessite pas un développement supplémentaire du réseau mobile pour pouvoir fonctionner. Cependant, de tels investissements pour le développement des applications coopératives et leur maintenance peuvent difficilement être rentabilisés pour les industriels. En effet, l'efficacité de ces technologies augmente avec leur adoption en masse, donc faire payer l'utilisation de ces applications peut en réduire l'efficacité.

La question de l'acceptabilité sociale de technologies de coopération dans un véhicule est également importante. En effet, on ressent une méfiance du grand public face à l'automatisation de certaines opérations dans la conduite. Le scénario d'un ordinateur incontrôlable qui se retourne contre les humains est d'ailleurs abondamment repris dans la fiction (HAL une intelligence artificielle, tue des astronautes dans *L'Odyssée de l'espace*), y compris dans le cas du contrôle des voitures (Dans la série Dr Who, ATMOS est un système anti-pollution véhiculaire capable de piloter les voitures à distance, et qui

tente d'éradiquer l'humanité). Par ailleurs, même en l'absence d'utilisateur malveillant, l'environnement juridique est encore incertain concernant des applications intégrées au véhicule, qui réagiraient à des données reçues de l'extérieur. Les responsabilités de l'expéditeur des données, du développeur de l'application, du constructeur automobile ou du conducteur pourraient éventuellement être engagées.

1.1.4 Déploiement réel de systèmes coopératifs

Les applications de coopérations véhiculaires constituent des applications réparties dans les véhicules en circulation. Dans un tel système réparti, la plupart des problèmes s'avèrent plus complexes à résoudre qu'avec une application centralisée. Les performances atteintes par les solutions réparties peuvent être très limitées, utilisant abondamment les ressources. Lorsque le système réparti est en mouvement, les déplacements des nœuds affectent aussi le résultat de l'application de coopération. En effet, en raison des changements de position, les véhicules ne peuvent disposer que d'une connaissance limitée de la topologie globale. Cela rend toute opération de routage très complexe, consommatrice de ressources, pour un résultat dont la durée de vie est limitée par les mouvements à venir des véhicules impliqués. La validation d'une application de coopération semble alors être une tâche difficile à mener, qui consomme beaucoup de ressources. Des déploiements expérimentaux à large échelle ont un coût dissuasif, tandis que des expériences de laboratoire, en environnement contrôlé sont trop peu représentatives de la réalité. En effet, le réseau routier est un réseau ouvert, où de nombreuses situations imprévues surviennent en permanence.

En raison du risque juridique et financier lié au déploiement d'une telle application véhiculaire de coopération, une application dont la validation n'est pas très solide ne pourra, en réalité, jamais être déployée. Ainsi, les applications de coopération véhiculaire ne bénéficient, à l'heure actuelle, toujours pas de déploiements opérationnels.

1.2 Contexte opérationnel de la thèse

Cette thèse a été menée au sein du laboratoire Heudiasyc (Heuristique et diagnostics des systèmes complexes), à Compiègne, dans les locaux de l'Université de Technologie de Compiègne (UTC). Le laboratoire Heudiasyc est une Unité Mixte de Recherche (UMR) associant le CNRS (Centre National de la Recherche Scientifique) et l'UTC. En raison de ses recherches, et afin d'en protéger le potentiel scientifique et technique, le laboratoire est soumis à une Zone à Régime Restrictif (ZRR), un dispositif visant à limiter les accès aux équipements et informations critiques ainsi qu'à assurer la sécurité des sites scientifiques à fort potentiel.

Le laboratoire accueille à la fois des chercheurs (liés au CNRS), des enseignants-chercheurs (liés à l'université) et des contractuels (doctorants, post-doctorants, etc.) qui se rassemblent par axe de recherche, afin d'évoluer dans un cadre commun, formant des équipes. L'appartenance à une équipe n'implique cependant pas le cloisonnement du personnel de recherche, qui est malgré tout souvent amené à coopérer avec les membres des autres équipes, par exemple dans le cadre de projets de recherche. Ces projets peuvent être financés par des entreprises ou des organismes publics, comme l'ANR (Agence Nationale de la Recherche), visant à orienter la recherche dans une direction qu'ils jugent prioritaire. Cette thèse est financée grâce au Labex MS2T (Maîtrise des Systèmes de Sys-

tèmes Technologiques), un projet transversal à différentes unités de recherches de l'UTC qui concerne l'étude des systèmes de systèmes technologiques. Les réseaux de véhicules en font partie, chaque véhicule constituant lui-même un système complexe. L'autre financeur de ce travail est l'AID (Agence Innovation Défense).

1.2.1 Équipes

Le personnel du laboratoire est regroupé au sein de 3 équipes distinctes qui conservent cependant de nombreuses interactions par des axes de recherche transversaux.

Connaissances, Incertitude, Données

L'équipe CID (Connaissances, Incertitude, Données) s'intéresse particulièrement au traitement de données, sous toutes leurs formes. Elle étudie l'information au moyen de méthodes de modélisation, mais également de calculs. L'intelligence artificielle fait partie intégrante de ses axes de recherche, à la fois par les méthodes symboliques et numériques, comme l'apprentissage ou les inférences, mais ses axes de recherche couvrent aussi l'interface avec les humains.

Systèmes Robotiques en Interaction

L'équipe SyRI (Systèmes Robotiques en Interaction) est tournée vers les systèmes embarqués, destinés à faire évoluer des robots dans un environnement ouvert. Ses recherches concernent notamment les drones et les voitures intelligentes. Elle étudie par exemple la perception embarquée, dans des robots ou dans une flotte de robots (perception collaborative). Elle s'intéresse également à la navigation autonome ou à la présence d'êtres humains parmi des robots.

Sûreté, Communication, OPtimisation

L'équipe SCOP (Sûreté, Communication, OPtimisation), s'intéresse à la conception, au déploiement, à la maintenance et à l'optimisation de systèmes sûrs et sécurisés. Elle étudie notamment la tolérance aux fautes et pannes, mais aussi aux attaques informatiques. Elle est également dédiée aux systèmes de communication, notamment les réseaux mobiles, mais aussi les réseaux de capteurs à énergie limitée.

C'est au sein de l'équipe SCOP que ce déroule cette thèse, sur la thématique des réseaux dynamiques. Cette thématique regroupe l'étude des algorithmes répartis et de leurs implémentations qui permettent de les mettre à l'épreuve empiriquement.

1.2.2 Plateformes expérimentales

L'expérimentation est au cœur de la démarche du laboratoire, qui propose à ses membres différentes plateformes expérimentales permettant une approche plus concrète de certains problèmes.

Une plateforme de réalité virtuelle permet notamment d'effectuer des expériences en environnement immersif simulé par ordinateur. Ces expériences permettent notamment d'étudier les interfaces homme-machine, dans des situations qui seraient autrement difficiles à reproduire dans la réalité. Elle permet également de simuler la conduite d'un train, dans laquelle coopèrent des systèmes automatiques et un conducteur.

Le laboratoire dispose également d'une plateforme dédiée aux mini-drones. Ces drones peuvent être testés en conditions réelles à l'intérieur de deux volières (dont une extérieure), qui permettent de faire voler de petites flottes de drones. Ces drones peuvent être équipés de perception, travailler en coopération ou en adversaires pour résoudre des tâches complexes.

Le laboratoire propose enfin plusieurs véhicules de tests, qui peuvent être équipés de capteurs et d'appareils embarqués. Certains véhicules sont entièrement robotisés, et peuvent être contrôlés par leurs systèmes embarqués, tandis que d'autres sont simplement équipés de systèmes qui ne peuvent agir sur le véhicule. Ces véhicules permettent d'effectuer des tests impliquant des véhicules en conduite autonome (grâce aux véhicules robotisés) et des véhicules classiques équipés de fonctionnalités intelligentes (qui pourraient s'intégrer à des systèmes d'aide à la conduite).

1.3 Contribution

Les applications de coopération véhiculaire ont été identifiées comme ayant un fort potentiel pour améliorer les performances du réseau routier, notamment pour prendre en compte des préoccupations de plus en plus importantes de la société civile. Malgré cela, leur déploiement ne semble pas se concrétiser sur les véhicules ou les infrastructures routières. Dans ce contexte, nous nous intéressons à la manière d'assurer la validation des applications de coopération entre véhicules au sein d'un réseau routier déjà complexe. Il a été identifié que la validation de ces applications pose problème, car elle est difficile et coûteuse à réaliser, tout en ayant du mal à être suffisamment solide pour ouvrir la voie aux déploiements réels. La contribution de ce travail de thèse est centrée sur l'élaboration d'une méthode de validation d'applications de coopération. Cette méthode doit faciliter le déploiement réel en étant simple d'utilisation et suffisamment solide. À cette fin, nous proposons une étude découpée en plusieurs étapes.

Analyse de la motivation et du besoin Afin d'élaborer une méthode de validation, notre étude s'intéressera aux validations déjà disponibles avant de déployer une application de coopération véhiculaire. Pour cela, un passage en revue des méthodes disponibles, qu'elles soient empiriques (basées sur des déploiements expérimentaux) ou théoriques (basées sur des analyses formelles) est nécessaire. Afin de mieux appréhender la validation empirique, l'étude de la diffusion fiable, un problème classique des réseaux sera conduite. Elle devra notamment permettre une étude expérimentale des performances de la solution proposée, qui mettra en lumière les enjeux d'une évaluation expérimentale d'application.

Élaboration une méthode de validation d'application de coopération Un autre défi que cette thèse vise à relever est l'élaboration d'une méthode de prédiction de performances. Cette méthode doit permettre de prédire les performances atteintes par une application de coopération véhiculaire exécutée au sein d'un réseau de véhicules. Son approche doit être suffisamment formelle pour limiter le besoin d'expériences de validation et suffisamment réaliste pour avoir une valeur prédictive convainquante.

Vérification de l'applicabilité de la méthode de prédiction proposée Le troisième défi que cette thèse vise à surmonter est de vérifier l'applicabilité de la méthode de prédiction

élaborée. Pour cela, différents scénarios véhiculaires réalistes devront être expérimentalement testés, et mis en lien avec la prédiction correspondante.

1.4 Organisation du manuscrit

La suite de ce manuscrit se présente comme suit :

Chapitre 2 Ce chapitre propose un état de l'art sur la coopération dans un réseau de véhicules afin de cerner plus précisément les contours du sujet. Il décrit les caractéristiques des réseaux de véhicules étudiés, illustre les applications possibles de la coopération et spécifie celles qui seront concernées par l'étude. Il s'intéresse également aux technologies sous-jacentes et aux moyens de valider une application de coopération véhiculaire.

Chapitre 3 Ce chapitre conduit, au sein d'un environnement véhiculaire, l'étude d'un problème classique des réseaux : la diffusion fiable. Il illustre les enjeux de la conception d'un algorithme réparti en réseau véhiculaire, ainsi que ceux d'une étude empirique de performance d'une application répartie implémentant cet algorithme.

Chapitre 4 Ce chapitre propose un état de l'art sur les modélisations permettant de rendre compte, à haut niveau, de la topologie d'un réseau dynamique. Ces modélisations permettent d'étudier formellement un algorithme réparti destiné à s'exécuter dans un réseau dynamique de véhicules. L'objectif de ce chapitre est de disposer des outils nécessaires à la conception d'une méthode de prédiction de performances d'algorithme réparti en réseau véhiculaire. Il propose également l'utilisation de certains des modèles pour démontrer l'algorithme de diffusion fiable proposé au chapitre 3.

Chapitre 5 Ce chapitre est consacré à l'élaboration d'une méthode de prédiction de performances d'un algorithme réparti en réseau véhiculaire. Cette méthode vise à simplifier la validation d'une application de coopération véhiculaire. Y est décrit le besoin d'une méthode de validation à la fois formelle et réaliste, puis les étapes de la méthode proposée.

Chapitre 6 Ce chapitre propose une évaluation empirique de la méthode de prédiction proposée au chapitre 5. Cette évaluation utilise des scénarios routiers simples et variés afin d'y appliquer la méthode de prédiction. Le résultat des expériences effectuées en laboratoire est utilisé pour caractériser l'efficacité de la méthode proposée.

Chapitre 7 Ce chapitre détaille l'étude, de bout en bout, de la résolution du problème de sécurité routière que constitue les collisions des véhicules avec les piétons. Il utilise des solutions issues de la littérature et des normes sur le sujet pour y appliquer la méthode de prédiction proposée au chapitre 5 dans des conditions issues d'une situation réelle. Les prédictions sont comparées à des résultats d'expériences de laboratoire et permettent d'illustrer l'application de la méthode dans un cas concret.

Conclusion Enfin, une conclusion générale et des perspectives de recherche sont proposées.

Chapitre 2

Coopération au sein d'un réseau dynamique de véhicules

Sommaire

2.1 Introduction

Les nouvelles technologies permettent l'émergence de réseaux véhiculaires, grâce auxquels les véhicules peuvent communiquer entre eux et avec l'infrastructure routière pour coopérer. L'intégration de fonctionnalités coopératives au sein d'applications véhiculaires est une tâche complexe, qui nécessite de bien comprendre les enjeux des technologies sous-jacentes dans le contexte véhiculaire. Afin de mieux cerner les contours de cette étude, nous nous intéressons tout d'abord à son contexte, c'est-à-dire comment des véhicules peuvent coopérer entre eux et avec l'infrastructure routière grâce aux nouvelles technologies, mais également comment cette coopération peut être étudiée. Dans ce chapitre, on s'intéresse tout d'abord aux enjeux de la coopération depuis l'application coopérative jusqu'au réseau véhiculaire. Nous détaillerons ensuite comment valider une application coopérative, à la fois d'un point de vue formel et expérimental.

2.2 Systèmes de Transport Intelligents Coopératifs (C-ITS)

L'utilisation de nouvelles technologies au sein d'un Système de Transport Intelligent Coopératif (C-ITS) permet d'améliorer les performances du réseau routier grâce à la

transmission d'information en son sein. Nous étudions tout d'abord les applications coopératives, avant de nous intéresser à l'architecture des communications sous-jacentes. Nous terminons cette section par une étude des technologies impliquées du point de vue des réseaux informatiques.

2.2.1 Applications coopératives dans un C-ITS

Les technologies de communication peuvent être intégrées aux véhicules par leur constructeur ou via des assistants de conduite, et aux éléments de l'infrastructure routière comme les caméras de surveillance, les feux tricolores. Les applications coopératives d'un C-ITS ont notamment le potentiel d'augmenter la sécurité routière en améliorant la connaissance des véhicules sur leur environnement. Les véhicules peuvent en effet se signaler mutuellement (communications de Véhicule à Véhicule V2V) leur présence grâce à des messages indiquant leur position et leur trajectoire future, comme le montre la figure 2.1 dans le cas d'une transmission simple des coordonnées GPS.

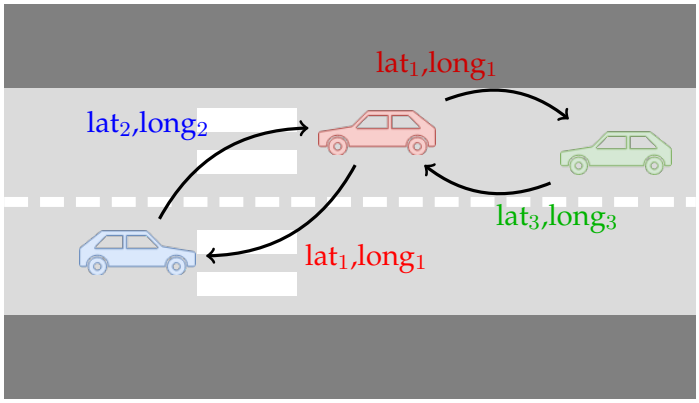


FIGURE 2.1 – Exemple d'utilisation de messages périodiques entre véhicules pour se signaler mutuellement leur position.

Grâce à ces messages, les différents véhicules peuvent facilement disposer d'une liste des véhicules proches en circulation, et ce même si ces derniers sont difficiles à repérer (zone de faible visibilité, conditions météorologiques). Cette connaissance peut permettre d'éviter les collisions, de la même manière qu'un radar anti-collision, tel que ceux équipant les Jaguar depuis 1999 [**?**], par exemple. Les véhicules peuvent également se transmettre via ces messages des informations sur leur état (allumage des feux) ou leurs intentions, comme la voiture bleue sur la figure 2.2, qui indique son intention de tourner à droite, de la même manière que les conducteurs utilisent leur clignotant.

Des applications coopératives pourraient également permettre de transmettre des alertes, comme par exemple lors d'un accident ou d'une panne, de la même manière que le triangle de signalisation ou les feux de détresse, mais avec une réactivité bien plus importante. C'est par exemple le cas du véhicule rouge, dans la figure 2.2, qui peut avertir le véhicule vert qu'il s'est arrêté dès l'arrêt complet voire à la fin du freinage. L'infrastructure peut également prévenir les véhicules de certains dangers (communications entre Véhicule et Infrastructure V2I), à la manière des gestionnaires d'autoroute qui préviennent via des panneaux d'affichage et une radio dédiée, tous de la même manière, certains événements dangereux, comme les véhicules à contresens [**?**], depuis 2015. Sur

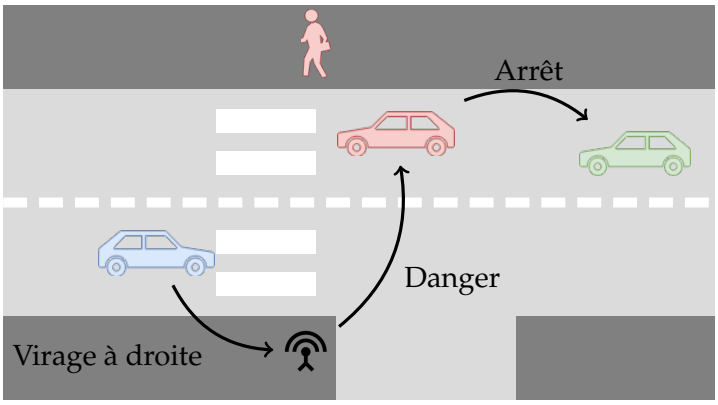


FIGURE 2.2 – Exemple de situation routière où des échanges entre des véhicules et l'infrastructure améliorent leur connaissance de la route.

la figure 2.2, la borne de bord de route (*Road Side Unit* ou *RSU*), qui a détecté un danger grâce à des capteurs de l'infrastructure, en prévient le véhicule rouge par l'envoi d'une alerte.

En dehors des applications liées à la sécurité routière, les C-ITS peuvent permettre une meilleure coordination des véhicules dans la circulation afin d'en optimiser le flux. On peut par exemple imaginer une régulation collaborative de vitesse sur autoroute adaptée à la densité du trafic. Les véhicules pourraient également collaborer pour se séparer sur deux itinéraires menant au même endroit (sur le modèle des itinéraires bis proposés par les gestionnaires de réseau routier) afin d'éviter qu'ils soient surchargés. D'autres usages sont également envisagés, pour informer les voyageurs, comme par exemple la transmission d'information sur les sites touristiques proches des routes, les lieux où réaliser des pauses ou des horaires de ferry. Il est enfin possible d'utiliser les fonctionnalités des C-ITS afin de divertir les passagers des véhicules, par exemple grâce à des jeux impliquant des passagers de différentes voitures. Dans le cadre de cette thèse, les applications de sécurité routière seront privilégiées, en raison des enjeux particuliers qu'elles concernent et des défis qu'elles présentent.

La multiplicité des acteurs qui peuvent être impliqués pose de nombreux problèmes, notamment en termes de confiance et de sécurité informatique [**?**]. En effet, si les assistants de conduite (voire des véhicules autonomes) réagissent aux données qu'ils reçoivent depuis l'extérieur, il est nécessaire de pouvoir filtrer des données non fiables, par exemple envoyées par un utilisateur malveillant ou issues d'échecs d'analyses voire de défaut de capteurs. Par ailleurs, les règles pour la protection des données personnelles des utilisateurs contraignent fortement les informations qui peuvent être communiquées. La conception du C-ITS doit en tenir compte, ce qui pousse la communauté à chercher des solutions d'authentification protégeant la vie privée des utilisateurs. Une stratégie commune pour cela est d'utiliser des pseudonymes, chaque véhicule est alors libre de changer régulièrement son identifiant parmi ceux dont il dispose. C'est l'approche proposée par [**?**], utilisant des pseudonymes temporaires tout en empêchant qu'il soit possible de faire le lien entre deux pseudonymes du même véhicule. Cette approche est basée sur des échanges avec l'infrastructure. Une autre stratégie de changement de pseudonyme est également proposée par [**?**] afin de gérer les pseudonymes utilisés par zone géographique. Une autre approche est proposée par [**?**], avec une architecture per-

mettant l'authentification anonyme de la source d'un message tout en assurant que la complexité des calculs engendrés reste raisonnable.

Les communications au sein du C-ITS n'ont pas toujours vocation à passer par le réseau Internet, notamment afin de limiter la latence des échanges qui a des impacts importants sur la sécurité routière. En effet, en se basant sur les communications WiFi, il est possible pour les véhicules de communiquer entre eux sans infrastructure prédéfinie (donc sans intermédiaire), par exemple via des transmission sans fil semblables au WiFi domestique. Cela permet également d'assurer le fonctionnement des applications de coopération véhiculaire même dans une zone blanche, voire dans un tunnel. Ainsi, alors que les communications sont nécessaires à tout système de transport coopératif, communiquer dans un environnement véhiculaire requiert à la fois des technologies réseau adaptées et des algorithmes de coopération qui prennent en compte les caractéristiques de l'environnement véhiculaire.

2.2.2 Architectures des communications

Dans un réseau de véhicules, une grande partie des machines du réseau sont embarquées dans des véhicules mobiles. Ainsi, le déploiement de technologies dédiées aux réseaux mobiles sans infrastructure prédéfinie (ad hoc) également appelés *Mobile Ad-hoc Networks* ou *MANET* peut sembler adapté pour gérer la coopération entre véhicules. C'est ainsi qu'on nomme VANET (*Vehicular Ad-hoc Netwok*) les réseaux de véhicules sans infrastructure prédéfinie, qui peuvent être considérés comme un cas particulier de MANET.

Les protocoles destinés aux MANET sont généralement destinés à supporter des mouvements épisodiques, dont la fréquence est plutôt faible, par rapport à la fréquence des échanges de données. En effet, ces protocoles cherchent à assurer un routage complet où chaque nœud mobile est capable de communiquer en unicast avec chaque autre nœud. Ils doivent pour cela mettre à jour et propager des tables de routage permettant de transmettre un paquet de bout en bout. C'est par exemple possible par des mises à jour périodiques (approche proactive) afin d'être prêt lors de la génération d'un paquet à transmettre comme OLSR (Optimized Link State Routing Protocol) [**?**] ou à travers une découverte de route réactive, lors de la génération du message à transmettre, comme c'est le cas de AODV (Ad-hoc On-demand Distance Vector) [**?**]. Ces protocoles de routage proposent des performances réduites lorsque la dynamique des nœuds du réseau est trop importante, car des tables de routage, parfois longues à établir et à propager, doivent être recalculées trop souvent [**?**, **?**].

Les véhicules, au contraire des nœuds mobiles des MANET, constituent un ensemble très hétérogène, où chaque machine est construite, maintenue et exploitée par des acteurs différents, et ont une mobilité moins aléatoire. L'hétérogénéité du réseau pose problème aux protocoles de MANET [**?**], tout comme le mouvement non uniformément aléatoire des véhicules du réseau. Leurs caractéristiques pourraient être exploitées pour élaborer un protocole de routage plus adapté aux VANET. Des recherches sur les MANET ont en effet déjà démontré qu'anticiper les ruptures de liens de communications peut permettre d'optimiser leurs protocoles de routages [**?**]. Le mouvement contraint des véhicules peut alors être mis à profit dans la conception d'un algorithme de routage. En effet, s'il est complexe de prédire la trajectoire complète d'un véhicule, son mouvement est contraint et peut ainsi être prédit, malgré une certaine incertitude [**?**] (les changements de trajectoire sont possibles mais ne peuvent pas être trop brusques, doivent suivre des routes,

leur sens de circulation). Une telle prédiction peut par exemple être réalisée grâce à des méthodes d'apprentissage profond comme dans [**?**], qui se base sur un historique récent de localisations. De telles prévisions de trajectoires peuvent permettre d'anticiper les ruptures de liens de communication qui peuvent être utilisés pour la conception de protocoles de routage spécifiques plus adaptés comme dans [**?**], qui propose un protocole de routage véhiculaire réactif basé sur la prédiction de positions.

Dans un VANET, il est cependant rare d'avoir une information à transmettre en unicast à un véhicule lointain. En effet, les applications de sécurité routière sont principalement intéressées par des données récentes et locales. Ces données peuvent être transmises d'un véhicule à l'autre, à travers une architecture de communication décentralisée nommée V2V (*Vehicle to vehicle*). Ces communications doivent se baser sur des transmissions par des véhicules en mouvement relatif parfois très important. En effet, sur autoroute, deux véhicules roulant à 130 km/h en sens opposé ne sont séparés par moins d'1 km que pendant environ 2 s.

Une architecture plus centralisée est possible lorsque l'infrastructure routière dispose d'équipements adaptés. Dans ce cas, les véhicules peuvent utiliser l'infrastructure routière fixe comme intermédiaire. C'est par exemple ce que propose [**?**] avec une application permettant d'éviter les collisions basée sur du MEC (*Multi-access Edge Computing*), c'est-à-dire l'utilisation des ressources de l'infrastructure proche des terminaux (dans ce cas, les RSU, proches des véhicules) pour exécuter l'application coopérative. L'infrastructure routière peut également jouer un rôle de source de données (transmission des limitations de vitesse, des travaux) ou de passerelle vers le réseau internet, comme le propose [**?**].

L'utilisation de communications combinant ces deux approches nommée V2X (*Vehicle to everything*) permet à la fois de bénéficier des performances de l'approche plus centralisée et de la résillience de l'approche décentralisée.

Grâce à ces architectures, les coopérations pour transmettre des données à la fois récentes et locales n'utilisent pas nécessairement de protocole de routage sur la totalité du réseau. La diffusion aveugle, c'est-à-dire l'émission en broadcast, à travers le protocole de communication sans fil, de la donnée sans connaître au préalable la topologie locale du réseau est dans ce cas suffisante et consomme moins de ressources et de temps qu'un routage (pas de découverte des routes). Lorsque la donnée reste pertinente pour des véhicules trop éloignés pour communiquer directement avec la source, une retransmission sur quelques sauts (éventuellement via l'infrastructure) peut être envisagée, et elle peut même être conditionnée aux paramètres de trajectoire des véhicules [**?**] afin d'en limiter la consommation de ressources tout en améliorant son efficacité.

2.2.3 Technologies réseau impliquées

Les protocoles de communication sans fil utilisés au sein d'un VANET doivent permettre des échanges sur des liens qui peuvent être très brefs (en raison du mouvement relatif des nœuds du réseau véhiculaire). La technologie de communication doit donc limiter au maximum les échanges protocolaires qui ne contiennent pas de charge utile. En effet, ces échanges diminuent davantage le « temps utile » du lien, et donc ses performances (notamment son débit).

Le WiFi domestique constitue une technologie de communication à courte portée capable d'assurer les transmissions d'un véhicule à un autre, comme le montrent par exemple les expériences menées dans [**?**]. Cependant, le protocole en mode infrastruc-

ture, tel qu'utilisé pour l'accès domestique à internet, impose une phase d'association centralisée entre les stations et le point d'accès, qui n'est pas très compatible avec les contraintes d'un réseau véhiculaire. Le mode ad hoc, utilisé dans les expériences véhiculaires évite cet écueil en décentralisant la phase d'association, et permet une communication directe, après l'établissement du lien. Cependant, la portée de ce système de communication est limitée à quelques dizaines de mètres, ce qui peut s'avérer faible en contexte véhiculaire, en particulier lorsque des vitesses importantes sont en jeu.

Pour assurer les communications au sein d'un VANET, le WiFi véhiculaire, une variation du WiFi domestique (802.11p) a été développé [**?**]. Il s'agit d'un protocole de communication utilisant la bande 5,9 GHz (en France). Il permet des échanges sans association entre les nœuds. Grâce à lui, des applications peuvent communiquer rapidement entre elles sur demande, mais aucun système n'existe pour assurer la fiabilité des transmissions ou l'identification des interlocuteurs. Ces fonctionnalités, si elles sont nécessaires doivent être déléguées à la couche applicative. Si la portée du WiFi véhiculaire est plus importante que celle du WiFi domestique, elle reste limitée à 1 km environ, ce qui permet aux liens de communication de supporter un certain éloignement relatif des véhicules au cours de la transmission tout en évitant d'interférer avec des nœuds trop éloignés.

Les technologies des réseaux mobiles 5G, actuellement en cours de déploiement, permettent d'utiliser des communications directes d'un terminal à un autre (D2D pour Device-to-Device) dont les caractéristiques se veulent similaires, avec une phase d'association courte mais centralisée par le réseau, permettant de laisser du temps aux communications utiles. Cependant, leur déploiement est encore embryonnaire en Europe et le mode D2D n'en bénéficie pas encore réellement.

Du fait de l'absence possible de phase d'association, les applications véhiculaires sont habituellement basées sur des émissions périodiques par tous les véhicules, souvent appelés messages BSM (Basic Safety Message) [**?**], en référence à une norme émise par le SAE (Society of American Engineer) pour l'Amérique du nord. Ces transmissions périodiques peuvent servir de support à d'autres communications, grâce à une fonctionnalité appelée piggy-backing. Elle consiste à éviter l'envoi d'un nouveau message lorsqu'une nouvelle donnée à transmettre apparaît pour la fusionner avec un message émis plus tard. Des informations supplémentaires ponctuelles peuvent ainsi être transmises lors de l'émission d'un message périodique. Cela permet de diminuer le nombre d'émissions (mais augmente la taille du message émis), afin notamment de limiter les interférences possibles avec d'autres nœuds du réseau.

Dans un VANET, les échanges sont ainsi basés sur des émissions périodiques effectuées par tous les véhicules sans connaissance de la topologie réseau. Le protocole réseau utilisé ne comprend généralement pas de phase d'association pour des questions de performances, et utilise des communications décentralisées, qui peuvent être directes entre les véhicules, ou impliquer l'infrastructure routière via des RSU, mais n'en sont pas dépendantes. Le déploiement progressif des réseaux mobiles 5G ont des applications en C-ITS grâce à leur fonctionnalité d'échanges directs entre terminaux, mais il s'agit d'un protocole centralisé disposant d'une phase d'association. Dans cet environnement, où la topologie change très vite sans que les nœuds du réseau ne puissent la connaître, la validation d'une application de coopération véhiculaire est à la fois une tâche complexe et cruciale pour son déploiement.

2.2.4 Normes sur les communications véhiculaires en Europe

La communauté des réseaux véhiculaires s'appuie sur des standards indispensables au fonctionnement d'un système coopératif. Ils sont établis par une collaboration de nombreux acteurs, non seulement au sein des organismes de normalisation comme l'ETSI (European Telecommunications Standards Institute) [**?**] en Europe, mais aussi au sein de projets de recherche comme le Car 2 Car Communication Consortium (C2C-CC) [**?**]. Ce dernier est un groupement de recherche européen qui implique des dizaines d'acteurs internationaux, comme Cohda Wireless, une entreprise australienne qui produit des solutions de communications destinées aux réseaux véhiculaires (V2X). L'objectif de ce consortium de recherche était de développer et tester des systèmes véhiculaires coopératifs. Il a permis des déploiements expérimentaux comme lors du European Truck Platooning Challenge 2016 [**?**] où plusieurs membres du C2C-CC ont fait participer un convoi équipé de systèmes collaboratifs.

Les industriels de ce consortium ont notamment participé à l'élaboration de certaines normes de communications de l'ETSI au niveau applicatif. Ainsi, le standard CAM [**?**] (Cooperative Awareness Message), qui désigne un message périodique (analogue au BSM) indiquant l'état du véhicule émetteur, avec, notamment sa position, sa vitesse, et des informations sur sa direction a été élaboré dès 2011, suivi par le standard DENM (Decentralized Event Notification Message), utilisé pour prévenir d'un événement affectant la circulation (alerte météorologique, fermeture de voie pour travaux, accident). La perception coopérative est supportée par le message normé CPM (Collective Perception Message) [**?**], introduit dès 2019 et capable de transmettre une liste d'objets avec leurs détails (position, vitesse, direction).

Ces normes considèrent une communication dédiée à courte portée (*Dedicated Short-Range Communication* ou DSRC), implémentée grâce à la norme 802.11p [**?**], qui définit le WiFi véhiculaire. Ce mode de communication est préféré aux communication via réseau téléphonique cellulaire (C-V2X) pour des questions de performances [**?**] et en raison du déploiement embryonnaire de la 5G en Europe lors du lancement du projet en 2008.

2.3 Validation d'une application de coopération véhiculaire

L'intégration d'une application de coopération véhiculaire au réseau routier nécessite de l'étudier, à la fois d'un point de vue formel à travers des modèles et par une méthode expérimentale. Nous examinerons tout d'abord comment une étude formelle peut permettre la démonstration d'un algorithme de coopération et la validation des applications qui l'implémentent. On s'intéressera ensuite à la validation expérimentale d'application de coopération véhiculaire.

2.3.1 Validation théorique d'applications véhiculaires coopératives

Il est possible d'étudier le fonctionnement d'une application véhiculaire par une étude formelle de son algorithme. Cette étude formelle vise à démontrer que l'algorithme réparti sous-jacent à l'application remplit ses objectifs. Il s'agit cependant d'une opération complexe, dont les difficultés limitent la portée des conclusions. Nous étudions tout d'abord les difficultés liées à la démonstration d'algorithmes répartis, puis l'impact des changements topologiques avant de nous intéresser au rapport à la réalité.

Difficultés de validation théorique des applications

La validation théorique d'algorithmes répartis est une méthode déjà éprouvée en réseau statique, qui permet d'offrir des garanties de bon fonctionnement de l'algorithme. Les démonstrations sont réalisées au sein d'un modèle de système réparti qui définit notamment comment les processus répartis communiquent entre eux, effectuent leurs calculs et comment des défaillances peuvent survenir dans le système.

Démontrer un algorithme réparti reste cependant une opération difficile, notamment lorsque l'algorithme ou le système étudié est complexe. Ainsi, certaines hypothèses, comme la présence d'un nombre suffisant de processus byzantins (des processus capables de se comporter de manière erratique à tout instant de l'exécution) peuvent facilement bloquer la résolution du problème, et donc empêcher de démontrer le bon fonctionnement de l'algorithme réparti dans ce système. Malgré cela, la présence de processus byzantins a des applications réelles, car elle permet de modéliser à la fois certaines pannes matérielles ou certains bugs logiciels et des utilisateurs malveillants, par exemple à l'origine de cyberattaques.

L'une des propriétés d'intérêt dans la démonstration d'un algorithme tolérant les défaillances est l'autostabilisation. Il s'agit d'assurer que l'algorithme finit toujours, même lorsqu'il est dans une configuration erronée (invalide), par atteindre une configuration valide. Cette propriété assure, même en présence de défaillances transitoires, que l'algorithme atteint ses spécifications si une période suffisamment longue sans défaillances a lieu au cours de l'exécution.

La topologie du réseau a également un impact sur les démonstrations. En effet, certaines démonstrations reposent sur des propriétés topologiques. C'est par exemple le cas de [**?**], dans lequel la démonstration d'un algorithme de calcul d'état est basée sur la structure topologique en anneau du réseau.

Les démonstrations d'algorithmes répartis sont ainsi souvent complexes à obtenir, elles font des hypothèses sur les communications, la topologie réseau, l'occurrence de défaillances et leur nature.

Impact d'une topologie dynamique

Lorsque le réseau a une topologie dynamique, il devient plus difficile de faire des hypothèses restrictives sur la topologie. Cela complique davantage le travail de démonstration. Par ailleurs, les variations de topologies peuvent générer des contraintes supplémentaires (par exemple la présence d'un changement topologique au cours d'une transmission).

Une démonstration est souvent contrainte de placer des restrictions sur les changements topologiques possibles. En effet, une topologie trop instable peut empêcher toute transmission de réussir et paralyser ainsi la coopération. Même lorsque la topologie est plus stable, des changements topologiques ayant lieu aux instants les plus cruciaux de l'exécution peuvent en provoquer l'échec.

Les contraintes sur les changements topologiques nécessitent d'utiliser un modèle de topologie dynamique. L'autostabilisation a par exemple fait l'objet de tentatives d'adaptation à une topologie dynamique. Cette approche est notamment celle de [**?**], où un algorithme de collecte est proposé puis démontrée. C'est également l'approche de [**?**], qui utilise un modèle dérivé des graphes (le modèle des TVG proposé par [**?**] et détaillé au chapitre 4 de ce manuscrit) pour étudier le problème d'élection de leader. Pour

chaque classe de graphe étudiée, un algorithme est conçu puis démontré à travers une adaptation de l'autostabilisation aux conditions de topologie dynamique. L'autostabilisation consiste alors à assurer que l'algorithme converge vers une configuration valide durant une période suffisamment longue sans changement topologique. La modélisation sous forme de p-graphes dynamiques (proposée par [**?**] et détaillée au chapitre 4) est utilisée par [**?**] pour la démonstration d'un algorithme de maintien de chemin en réseau véhiculaire malgré des changements topologiques.

Dans les démonstrations d'algorithmes répartis en réseau dynamique, la modélisation de certaines défaillances est généralement simplifiée. En effet, les pertes de messages peuvent être modélisées par un changement de topologie déconnectant temporairement l'émetteur et le récepteur du message au cours de sa transmission.

Réalisme des hypothèses

Les démonstrations d'algorithmes répartis en réseau dynamique sont donc basées sur des hypothèses simplificatrices de la réalité, explicitées à la fois dans la modélisation du système réparti et dans la modélisation des changements topologiques. En raison des difficultés de démonstration des algorithmes répartis dans une topologie dynamique, ces hypothèses sont généralement choisies très restrictives. C'est par exemple le cas dans [**?**, **?**], où les auteurs supposent la récurrence d'arêtes au sein d'un anneau dynamique.

Des hypothèses très restrictives facilitent le travail de démonstration mais peuvent impacter le réalisme des modélisations utilisées, et rendre moins réalistes les conclusions de la démonstration ou réduire leur portée. Dans [**?**], les auteurs supposent que les nœuds du réseau sont capables d'identifier instantanément les changements topologiques. Cette hypothèse est assez peu réaliste, mais peut représenter une situation dans laquelle des émissions périodiques ont lieu et permettent de rapidement détecter ses voisins. Cependant, de telles émissions périodiques ne sont généralement pas sans effet sur le bon déroulement de l'algorithme (interférences avec les émissions de l'algorithme, consommation de ressources réseau, voire d'énergie), ce dont la modélisation ne tient pas compte.

2.3.2 Validation expérimentale d'application véhiculaire coopérative

Une étude empirique de l'application de coopération permet de vérifier son fonctionnement et de mesurer ses performances. Cependant, une telle étude n'est pas facile à réaliser, et nécessite des outils et des simplifications pour parvenir à des conclusions exploitables. Nous étudions tout d'abord les développements préalables à toute expérimentation, les outils disponibles, puis les choix effectués pour le travail de cette thèse.

Développements réels

Tout d'abord, expérimenter sur une application de coopération véhiculaire nécessite de disposer d'une telle application. Une étape de développement est donc nécessaire pour implémenter l'algorithme réparti qui peut être étudié formellement en une application répartie qui peut être réellement exécutée au cours de tests.

Au laboratoire Heudiasyc, certains outils facilitent le développement de telles applications réparties destinées à être utilisées au sein d'un réseau dynamique comme un

réseau de véhicules. La suite logicielle Airplug [**?**, **?**] constitue ainsi un intergiciel (*middleware*) facilitant la communication entre différents processus s'exécutant sur un même appareil ou sur des machines différentes.

L'environnement Airplug considère que chaque instance d'une application répartie est un processus à part entière, qui communique avec les autres instances et avec d'autres applications réparties par ses descripteurs de fichiers. Un format simple et polyvalent de message permet à chaque application d'interpréter les informations qu'elle reçoit.

Le développement d'une application compatible avec les spécifications Airplug est facilité par l'existence de bibliothèques en différents langages (TCL, C, Python) permettant d'implémenter rapidement un algorithme réparti capable d'interagir avec un utilisateur tout comme d'autres applications réparties.

Outils pour les expérimentations

Il est habituel que chaque équipe de recherche utilise ses propres outils pour ses études expérimentales, afin de faciliter le développement et l'interfaçage avec les différents composants matériels et logiciels de l'expérience. En effet, le déroulement de l'expérience est fortement dépendant de choix techniques comme celui des véhicules, qui peuvent être de véritables voitures équipées d'ordinateurs embarqués ou des robots miniatures. Les dispositifs de communication et leur interfaçage avec les systèmes embarqués dépendent des besoins des expérimentateurs ce qui aboutit à des architectures expérimentales très différentes d'une équipe de recherche à l'autre.

Lorsque les expériences sont virtuellement réalisées sur ordinateur, il est possible d'utiliser un simulateur de trafic routier comme SUMO [**?**] afin de générer un trafic réaliste. Le comportement coopératif des véhicules peut être modélisé grâce à un simulateur d'événements discrets qui calcule le comportement des communications comme par exemple NS [**?**], OMNET++ [**?**] ou the ONE [**?**].

Émulateur utilisé

Lorsque les expériences doivent être nombreuses et concerner des véhicules plus nombreux que ce que le laboratoire peut offrir, la suite logicielle Airplug propose un émulateur de réseaux dynamiques qui a démontré un certain degré de réalisme [**?**]. Il permet d'utiliser une application Airplug (la même que pour des expériences réelles) en virtualisant lui-même les communications entre les nœuds. Les communications entre applications sont assurées par des pipes, gérés dynamiquement par l'émulateur selon un scénario préétabli. L'émulateur peut prendre en compte des trajectoires GPS captées sur des véhicules réels ou générées (par exemple par un simulateur de trafic) et afficher les déplacements sur une carte OpenStreetmap [**?**]. Les paramètres techniques (portée du système de communication, latence) peuvent être configurés dans l'émulateur tout comme les paramètres environnementaux (taux de pertes de messages, atténuation du signal). Les expériences en émulation, même si elles ont lieu dans un environnement simplifié et contrôlé, permettent d'effectuer facilement des tests en réseau véhiculaire avec des applications réparties sans trop sacrifier le réalisme des résultats.

Architecture d'expérimentations réelles

Toute expérimentation réelle nécessite un environnement matériel adapté. Dans le cas d'expérimentation sur des applications véhiculaires, du matériel est nécessaire : des véhicules, mais surtout du matériel de communication. Le laboratoire Heudiasyc dispose de plusieurs véhicules expérimentaux, qui peuvent être équipés de transmetteurs utilisant le WiFi véhiculaire (802.11p) de la marque Cohda Wireless. Les applications réparties peuvent être exécutées sur des cartes Raspberry Pi interfacées par ethernet avec les transmetteurs. Alternativement, il est possible d'utiliser une carte WiFi externe (USB), reliée à une antenne externe placée sur le toit du véhicule. La carte WiFi doit alors être configurée en mode ad hoc. Cette architecture matérielle peut également être utilisée en position fixe pour former des bornes de l'infrastructure routière (RSU).

Il est également possible d'utiliser des cartes Raspberry Pi intégrées directement à des robots miniatures. Ces derniers peuvent interagir entre eux directement via leur carte WiFi, configurée en mode ad hoc. Compte tenu des distances réduites entre les robots miniatures, l'utilisation d'une antenne externe n'est alors pas nécessaire.

Lors d'expériences réelles (utilisant des véhicules ou des robots miniatures), Airplug assure l'interface avec le réseau [**?**] ce qui rend, pour l'application répartie, les interactions avec d'autres instances ou d'autres applications transparentes et indépendantes des technologies de communication utilisées. Cependant, de telles expériences requièrent de nombreuses ressources logistiques et ne sont pas entièrement automatisables. Elles ne sont ainsi réalisables qu'en petit nombre et il est alors difficile d'établir leur représentativité.

2.4 Conclusion et positionnement

À l'aide des notions extraites de la littérature sur les C-ITS et la validation d'applications coopératives dans un réseau véhiculaire, nous proposons de détailler un peu plus le positionnement de l'approche engagée dans ce manuscrit. Nous décrivons à cette fin les outils et suppositions utilisés dans le cadre de l'étude d'applications de coopération véhiculaires en vue de leur validation.

Dans le cadre de ce manuscrit, certaines hypothèses sont posées afin de délimiter clairement le cadre dans lequel s'inscrit cette étude visant à permettre la validation d'applications de coopération véhiculaire. Nous définissons ici les caractéristiques du réseau dans lequel les applications de coopération étudiées sont destinées à être exécutées.

Le réseau véhiculaire est ainsi considéré comme ne disposant pas de protocole de routage général, même si les véhicules possèdent systématiquement d'un identifiant leur permettant de se différencier les uns des autres. La présence d'une connexion à internet fiable dans les véhicules n'est pas supposée, pour prendre en compte les zones blanches et limiter la latence et l'usage de ressources.

L'étude d'une exécution d'application répartie par émulation, avec les outils du laboratoire Heudiasyc, est, par ailleurs, considérée représentative de la réalité, conformément aux résultats de tests réels réalisés précédemment [**?**]. Cependant, l'étude d'une exécution en émulation n'est pas considérée prédictive, ce qui signifie que plusieurs émulations du même scénario peuvent, selon des paramètres qu'il est difficile de maîtriser dans la réalité, aboutir à des résultats différents.

Maintenant que le cadre de l'étude menée au cours de cette thèse est mieux défini,

il devient possible d'utiliser le problème de la diffusion fiable pour concevoir et valider empiriquement un algorithme qui le résout dans un réseau dynamique de véhicules.

Chapitre 3

Diffusion fiable coopérative : Algorithme et validation

Sommaire

3.1 Introduction

Dans les réseaux dynamiques de véhicules, toute opération basique devient rapidement complexe, et la conception de solutions à des problèmes classiques des systèmes

répartis peut s'avérer fastidieuse. Elle est cependant nécessaire, pour valider les applications de coopération destinées à améliorer la fluidité du trafic ou la sécurité routière.

Dès lors, la conception des algorithmes destinés aux réseaux de véhicules devient un enjeu crucial, tout comme l'évaluation de leurs performances. Parmi les problèmes classiques étudiés dans les réseaux informatiques, la diffusion fiable consiste à distribuer une donnée à tous les nœuds membres du réseau.

Dans ce chapitre, une description du problème de la diffusion fiable en contexte véhiculaire est tout d'abord proposée, suivie des approches de la littérature, avant la conception d'un algorithme qui le résout. Une étude de performance empirique est finalement conduite sur cet algorithme.

3.2 Problème de la diffusion fiable

Dans la plupart des réseaux disposant de protocoles de contrôle, diffuser un message à tous les membres du réseau est une primitive de base sur laquelle peuvent reposer de nombreux algorithmes répartis (réservation de ressources, consensus, élection).

Malgré cela, dans un réseau ad hoc ne disposant pas d'un protocole de routage et dont les nœuds sont en mouvement, assurer une diffusion fiable reste une tâche complexe. L'étude de ce problème permet d'aborder sous plusieurs angles les enjeux des algorithmes répartis en réseaux véhiculaires.

3.2.1 Description du problème

La diffusion d'une donnée consiste à envoyer cette donnée à tous les membres d'un réseau. Il s'agit d'une action basique dans tout réseau, parfois à la base d'applications plus complexes (par exemple l'allocation d'identifiants ou le routage). Cependant, puisque les réseaux (en particulier les réseaux sans fil) ne sont pas parfaits, il est possible qu'un des nœuds ne reçoive pas l'information à diffuser, et ce même si elle lui a été envoyée.

Effectuer une diffusion fiable consiste à s'assurer que tous les nœuds du réseau ont bien reçu l'information diffusée, même si certaines transmissions échouent. Cela implique de retransmettre le message et, par conséquent, de détecter les nœuds n'ayant pas reçu l'information.

Afin de pouvoir renvoyer la donnée à chacun des nœuds qui ne l'ont pas reçue, il est nécessaire de garder en mémoire la donnée à diffuser sur au moins l'un des nœuds du réseau. Dans la mesure où les ressources de mémoire sont limitées, il est également primordial de supprimer de la mémoire les messages dont la diffusion est complète.

Comme indiqué au chapitre 2, les nœuds du réseau sont supposés disposer d'un identifiant unique, et communiquent via une technologie sans fil sans connaître la topologie réseau ni même leurs voisins.

3.2.2 Difficultés liées au réseau véhiculaire

Dans un réseau véhiculaire, les nœuds sont en mouvement. Ainsi, lors de la réception d'un message émis par véhicule proche, un nœud ne peut jamais être certain de pouvoir répondre à l'émetteur du message qui pourrait déjà ne plus être à portée de communication. Dans un réseau véhiculaire, même en l'absence de pertes de messages, il reste possible qu'une diffusion soit incomplète. En effet, un véhicule qui se déplacerait d'une

zone où aucun de ses voisins n'a reçu la donnée vers une zone où tous ses voisins ont déjà arrêté de la transmettre serait incapable de la recevoir. Les risques d'une diffusion incomplète en sont donc largement augmentés.

Les ressources réseau dans un contexte véhiculaire sont partagées entre différents nœuds, afin qu'ils puissent communiquer avec tous leurs voisins. En effet, comme les véhicules ne savent pas à l'avance quels sont leurs voisins ni à qui ils s'adressent, il leur est difficile de se coordonner pour utiliser des canaux différents et donc de limiter l'impact des interférences électromagnétiques.

Les équipements utilisés au sein d'un réseau véhiculaire sont tous des équipements embarqués, qui disposent de capacités de calcul limitées ainsi que leurs ressources mémoire. Ces ressources peuvent par ailleurs être mobilisées pour de nombreuses et diverses tâches (traitement des capteurs, communications, interface avec le conducteur et ses passagers...).

Pour ces raisons, des algorithmes très gourmands en capacités de calcul ou en mémoire semblent inadaptés au contexte véhiculaire.

3.2.3 Applications dans le domaine routier

Dans la plupart des applications véhiculaires, les informations doivent transiter rapidement d'un véhicule à l'autre. Cependant, certains algorithmes de coopération requièrent que tous les véhicules y participant aient reçu le résultat. C'est par exemple le cas des algorithmes répartis de réservation de ressources. Dans un contexte véhiculaire, la réservation de ressources peut servir au franchissement d'intersection ou à la gestion des dépassements [**?**].

Pour assurer ce type de fonctionnalité, on utilise généralement une approche centralisée parce que son implémentation est plus simple. Cependant, un algorithme décentralisé permettrait de se passer d'infrastructure plus facilement et donc de faciliter le déploiement. Un algorithme efficace de diffusion fiable peut permettre la réservation décentralisée de ressources.

Dans un convoi de véhicules, la prise de décision collaborative peut également nécessiter une bonne diffusion des informations à tous les membres du convoi. C'est par exemple le cas au sein d'un convoi, lorsqu'est définie, via son leader, une vitesse de consigne ou une direction à prendre.

La diffusion fiable est également un problème rencontré lors des diffusions d'alertes, comme par exemple dans [**?**] qui propose un protocole de diffusion d'alerte disposant de mécanismes de fiabilité assurant une prise en compte du danger par un nombre maximal de véhicule.

3.3 État de l'art

Les problèmes de diffusion dans les réseaux véhiculaires sont traités de différentes manières selon les objectifs et les contraintes applicatives. Nous proposons tout d'abord d'étudier les solutions les plus simples, qui considèrent qu'un seul nœud peut être source de diffusions, avant de nous intéresser aux études qui prennent en compte les défaillances byzantines. Les solutions faisant l'impasse sur la fiabilité seront enfin présentées.

3.3.1 Approche de diffusion à source unique

L'approche la plus intuitive pour réaliser une diffusion fiable depuis une unique source pourrait être de demander des accusés de réception aux autres nœuds, permettant de s'assurer de la bonne réception de la donnée par tous. Par ailleurs, le moyen le plus simple d'arriver à une diffusion dans un réseau dont on ignore la topologie est une propagation de la donnée selon un arbre couvrant tout le réseau [**?**]. Ce type de propagation permet en effet de limiter le nombre d'envois et donc de limiter l'usage de la bande passante. De plus, il permet de rejoindre tous les nœuds avec un nombre minimal de saut et donc une latence réduite tout en évitant le risque de boucle réseau pouvant mener à une tempête de diffusion (*broadcast storm*). Le système le plus simple assurant une diffusion fiable pourrait donc être composé d'une propagation initiale suivant un arbre couvrant où les nœuds accusent réception de la donnée en remontant à la racine de l'arbre comme proposé dans [**?**].

Dans un réseau sans fil, les nœuds ne peuvent pas directement savoir s'ils sont les feuilles de l'arbre couvrant. Ils doivent ainsi tous relayer le message à diffuser une fois (en raison de la propagation radio, tous les nœuds à portée pourront le recevoir s'il n'y a pas de perte aléatoire). De la même manière, il est impossible pour un nœud de savoir s'il a reçu des accusés de réception de la part de tous ses nœuds fils avant de renvoyer son propre accusé. C'est pourquoi chaque nœud doit attendre un certain délai, proportionnel à sa profondeur dans l'arbre couvrant. Après un certain délai dépendant de la profondeur maximale, s'il manque des accusés de réception, la diffusion pourrait repartir. C'est en substance le fonctionnement de l'algorithme DPIF présenté dans [**?**] qu'il faudrait relancer tant qu'il manque des accusés de réception. Cependant, un tel algorithme nécessite beaucoup de ressources (tant que la diffusion est incomplète, une nouvelle diffusion totale est lancée, impliquant l'émission de nombreux messages inutiles, tout comme de nombreux accusés de réception déjà collectés. Si plusieurs nœuds servent de source, le réseau peut vite être saturé par les émissions.

3.3.2 Défaillances byzantines

Dans l'étude des systèmes répartis, de nombreux travaux s'intéressent à la gestion des défaillances byzantines telles que définies dans [**?**]. Ces défaillances sont très difficiles à identifier et pallier. En effet, dans le modèle byzantin, certains nœuds dits malicieux sont capables d'agir selon leur gré au lieu de suivre les algorithmes prévus. Il leur est entre autres possible d'envoyer des messages non justifiés par l'algorithme, de ne pas envoyer des messages requis par l'algorithme ou encore de s'arrêter complètement de fonctionner.

Ce modèle permet à la fois de représenter des attaques protocolaires volontaires et des défaillances accidentelles. Un attaquant utilisant les propriétés du protocole ne pourrait en effet réaliser que des actions similaires à celles d'un nœud byzantin malicieux. Les défaillances accidentelles, quant à elles sont généralement plus limitées que les défaillances byzantines mais il reste possible, par exemple, que des capteurs défaillants ou un bug logiciel conduisent l'algorithme à calculer des valeurs erronées sur un nœud qui faussent le contenu de ses messages.

Dans un modèle de défaillances byzantin, le moindre problème devient impossible à résoudre si le nombre de nœuds byzantins n'est pas limité. On peut facilement s'en convaincre en prenant un exemple où tous les nœuds sont byzantins. On s'attache alors

à concevoir des algorithmes capables de supporter un maximum de nœuds byzantins dans des conditions précises (topologie réseau, durée de l'algorithme).

Un algorithme proposé par [**?**] cherche à assurer un consensus malgré la présence de nœuds malicieux byzantins. Un consensus ne peut avoir lieu sur une valeur que si tous les nœuds ont reçu ladite valeur, donc si une diffusion fiable de la valeur a eu lieu. C'est pourquoi la solution proposée, dans sa version concernant les réseaux à topologie inconnue commence par proposer un algorithme de diffusion. Dans cette solution, un message contient la route qu'il a empruntée depuis sa source. L'identifiant de chaque relai est ajouté au message par le récepteur du message. Ainsi, même si le relai est un nœud byzantin, il ne peut altérer lui-même sa participation à la route. Le récepteur analyse alors tous les messages qu'il a reçus et cherche ceux qui ont emprunté une route entièrement différente (pas de relais communs). Chaque route peut avoir été altérée par un ou plusieurs nœuds byzantins. Cependant, le dernier nœud byzantin de la route ne peut être caché. Dans ces circonstances, la réception d'un même message par plus de routes entièrement différentes qu'il n'y a de nœuds byzantins assure qu'au moins l'un de ces messages est passé par une route sans aucun byzantin, et est, par conséquent fiable.

Une adaptation du protocole [**?**] est proposée dans [**?**] pour tenir compte des caractéristiques d'un réseau dynamique et chercher des conditions sur la topologie locale des nœuds. Pour prendre en compte les changements topologiques, chaque nœud renvoie son état dès qu'un changement topologique a lieu. Cet algorithme repose donc sur une connaissance de la topologie, ce qui ne peut être obtenu, dans un réseau véhiculaire, que grâce à un autre algorithme réparti impliquant une consommation de ressources supplémentaires. Une approche différente est proposée par [**?**], avec un algorithme destiné à des réseaux sans fil statiques. Du fait des pertes de messages, on peut considérer un tel réseau comme dynamique, puisque certains liens peuvent temporairement disparaître et réapparaître. L'approche considérée se base sur une série d'élections de leaders locaux connectés entre eux, qui serviront de structure de contrôle au réseau.

Ces approches sont difficiles à utiliser dans un réseau véhiculaire car elles reposent sur la redondance de transmission des messages à l'intérieur du réseau de véhicules, ce qui se traduit par une forte consommation de ressources réseau. Par ailleurs, l'authentification des véhicules sur le réseau pourrait le protéger des nœuds malveillants avec une consommation de ressources réseau plus faible. Quant aux défaillances accidentelles, elles sont habituellement moins graves que des défaillances byzantines et pourraient être étudiées avec d'autres modèles de système réparti. Des défaillances byzantines accidentelles peuvent malgré tout survenir en raison de défauts matériels (panne de capteurs) ou logiciels (bug), et des stratégies pour les résoudre s'avéreront malgré tout nécessaires.

3.3.3 Fiabilité non garantie

D'autres approches de la diffusion préfèrent limiter la fiabilité de la diffusion, généralement dans le but d'améliorer les performances, notamment la consommation de ressources et la durée de diffusion. C'est le cas de PGB (Preferred Group Broadcast), un protocole proposé par [**?**], basé sur la constitution de groupes en fonction de la puissance du signal reçu. L'objectif est de limiter la durée de diffusion et d'éviter les messages redondants tout en assurant qu'un lien ne soit pas interrompu pendant la transmission.

Chaque nœud du réseau trie ses voisins en fonction de leur distance ou de la puissance du signal reçu. Il les classe en 3 groupes : les voisins trop proches (ou dont le

signal est trop élevé), le groupe préféré, et les voisins trop loin (ou dont le signal est trop faible). À la réception d'un message qui doit être retransmis, le receveur choisit un délai d'attente dépendant du groupe auquel il appartient. Il attend ce délai avant de retransmettre. Pour améliorer la fiabilité, l'émetteur de chaque message écoute le réseau après avoir transmis un message. S'il ne reçoit pas de retransmission, il répétera l'émission à nouveau. Ce mécanisme n'assure pas que le message atteindra tous les nœuds, mais il permet d'éviter une rupture de retransmission affectant une zone entière.

Une autre approche est proposée par [**?**] sous le nom de CPB (*Clustering and Probabilistic Broadcasting*), un protocole basé sur la formation de groupes stables, au sein desquels les un leader est choisi. C'est ce leader qui a la charge de relayer l'information, mais les autres nœuds peuvent le faire (avec une certaine probabilité dépendant de la densité du réseau) s'ils jugent que le leader n'est pas en mesure d'accomplir cette tâche (s'il est trop loin de l'émetteur). Une approche similaire basée sur les MPR (Multi-Point Relay) est proposée par [**?**], avec une contrainte supplémentaire sur l'énergie dépensée par les différents nœuds du réseau.

Dans un réseau véhiculaire, lorsque la fiabilité n'est pas essentielle, il est possible d'obtenir de meilleures performances (notamment sur l'utilisation réseau et la durée de transmission) en sacrifiant partiellement la fiabilité. Lorsque la fiabilité est une fonctionnalité non sacrifiable, il faut concevoir un algorithme capable d'assurer une diffusion fiable au sein d'un réseau véhiculaire dynamique. On choisit à cette fin de ne pas se baser sur la présence d'un protocole de routage, qui s'avère à la fois complexe et pas toujours nécessaire dans un réseau véhiculaire. Les défaillances byzantines ne seront pas considérées, afin de limiter l'usage des ressources.

3.4 Algorithme de diffusion fiable véhiculaire RDF

Nous décrivons dans cette section la proposition originale d'un algorithme nommé RDF (Reliable DifFusion [**?**, **?**]) répondant aux caractéristiques définies dans la section 3.3. Cet algorithme est progressivement construit dans cette section, en commençant par le mécanisme de diffusion. Nous détaillons ensuite la manière de détecter qu'un nœud du réseau a manqué une diffusion avant de nous intéresser à la vidange des messages correctement diffusés de la mémoire.

3.4.1 Diffusion d'information

La première étape de l'algorithme de diffusion fiable RDF est d'assurer une diffusion du message sans avoir à faire d'hypothèse restrictive sur la topologie du réseau. À cette fin, l'algorithme est basé sur une adaptation de l'algorithme PI [**?**] où chaque nœud transfère une unique fois à tous ses voisins le message à diffuser lorsqu'il le reçoit. Le fonctionnement de cette étape est décrit par la figure 3.1.

Chaque diffusion est identifiée par le couple composé de l'identifiant de la source de la diffusion et du numéro de séquence (local à la source) de la diffusion. L'initiateur de la diffusion envoie un message à ses voisins comprenant l'information à diffuser, mais aussi l'identifiant de cette diffusion. À réception d'un message, le récepteur vérifie s'il a déjà participé à cette diffusion grâce à l'identifiant, et le retransmet si tel n'est pas le cas. L'algorithme 1 détaille une version intermédiaire de l'algorithme, qui décrit comment cet algorithme diffuse un message.

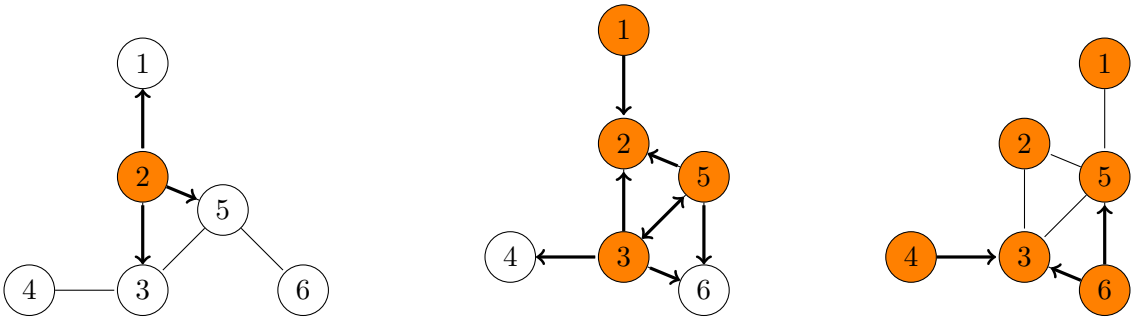


FIGURE 3.1 – Propagation initiale de l'information au cours du temps, qui s'écoule de gauche à droite. Le nœud 2 démarre la diffusion et transmet à ses voisins l'information, puis ses voisins reçoivent l'information et la retransmettent à leur tour. À la fin, tous les nœuds du réseau ont reçu l'information malgré de rares changements topologiques.

Le réseau est supposé composé de N nœuds différents ayant chacun un identifiant unique compris entre 1 et N. Chaque nœud du réseau connaît la valeur de N et son propre identifiant. À réception d'une nouvelle diffusion, le récepteur (d'identifiant *id*) mémorise (dans la liste d'association $C_{id}[src]$) la donnée diffusée. Lorsque toutes les diffusions d'un nœud d'identifiant i jusqu'à la diffusion n ont été reçues, il est possible de compresser cette information en ne retenant que le numéro du dernier message de la série consécutive. Cette opération peut être réalisée pour chaque initiateur i et son résultat peut être trouvé dans l'élément $M_{id}[id][i]$ de la matrice M_{id}.

Algorithme 1 : **RDF - étape 1 - diffusion *id***

Initialisation :
  pour chaque $i \in \{1, \ldots, N\}$ **faire**
    pour chaque $j \in \{1, \ldots, N\}$ **faire**
      ▷ $M_{id}[l][m]$ *représente la connaissance par le nœud id du dernier message généré sur le nœud* l *que le nœud* m *a reçu.*
      $M_{id}[i][j] \leftarrow 0$
    fin pour
    ▷ $C_{id}[l]$ *est un dictionnaire des messages reçus par le nœud id, générés sur le nœud* l*.* $\{\}$ *est un dictionnaire vide.*
    $C_{id}[i] \leftarrow \{\}$
  fin pour
  ▷ *Liste des messages à envoyer, traitée par le protocole d'envoi de messages*
  $file\_envois_{id} \leftarrow []$

Nouvelle diffusion localement déclenchée :
  ▷ *La donnée payload doit être diffusé depuis le nœud id vers tous les autres nœuds*
  $M_{id}[id][id] \leftarrow M_{id}[id][id] + 1$
  $nseq \leftarrow M_{id}[id][id]$
  $C_{id}[id][nseq] \leftarrow payload$
  append($file\_envois_{id}$,$(id, nseq)$)

Réception d'un message RDF :
  recevoir(*message* **)**

▷ *identifiant du nœud ayant généré le message*
16 $s \leftarrow$ *message*.source
▷ *numéro de séquene du message*
17 $n \leftarrow$ *message*.nseq
18 *pld* $\leftarrow m$.payload
▷ *Faire suivre le message si le message n'a pas encore été diffusé par le nœud id*
19 **si** $n > M_{id}[s][id]$ **et** $n \notin C_{id}[s]$ **alors**
▷ *Sauvegarde en mémoire du message reçu*
20 $C_{id}[s][n] \leftarrow$ *pld*
▷ *Transfert du message pour poursuivre la propagation*
21 append($file\_envois_{id}(s, n)$)
22 **fin si**
▷ *Mise à jour de la matrice avec le numéro du dernier message (q) généré par le nœud s dont tous les prédécesseurs sont reçus*
23 $q \leftarrow n$
24 **tant que** $q + 1 \in C_{id}[s]$ **faire**
25 $q \leftarrow q + 1$
26 **fin tant que**
27 $M_{id}[s][id] \leftarrow q$

28 Expiration du timer d'envoi :
29 **si** $file\_envois_{id} \neq \emptyset$ **alors**
30 $(s, n) \leftarrow$ pop($file\_envois_{id}$) ▷ *identifiant du message à envoyer*
31 $m \leftarrow$ nouveau_message()
32 m.source $\leftarrow s$
33 m.nseq $\leftarrow n$
34 m.payload $\leftarrow C_{id}[s][n]$
35 **envoyer(** m **)**
36 **fin si**
37 Réarmer le timer d'envoi

3.4.2 Détection d'un message manqué

Chaque nœud du réseau, peut, lorsqu'il émet un message, y ajouter des informations sur son propre état. Ces informations peuvent permettre aux nœuds qui recevront le message de détecter s'il a manqué des diffusions en cours. En effet, il est possible pour l'émetteur d'un message *rel*, d'envoyer les identifiants des diffusions auxquelles il a bien participé. Grâce à cela, un véhicule voisin pourrait décider d'envoyer un message dont il dispose en mémoire et qui a été manqué par *rel*, comme le montre la figure 3.2. Il serait alors en mesure de le lui renvoyer afin de faire progresser le nœud *rel*. En y ajoutant ce mécanisme, RDF prend la forme intermédiaire décrite à l'algorithme 2.

Pour limiter la taille du message, seule la ligne $M_{rel}[rel]$ est envoyée, car elle contient en un espace réduit (un entier par nœud du réseau) des informations concernant un grand nombre de diffusions reçues par le nœud *rel* initiées par tous les nœuds du réseau. La détection d'un voisin retardataire se fait en comparant l'état qu'il transmet dans son message (V_{rel}) avec l'état du nœud receveur ($M_{id}[id]$). Si le récepteur dispose, pour l'un des nœuds du réseau, d'une diffusion plus récente que l'émetteur du message (si $\exists q; V_{rel}[q] < M_{id}[q][id]$), il l'ajoute aux messages à retransmettre en priorité. En effet, s'il

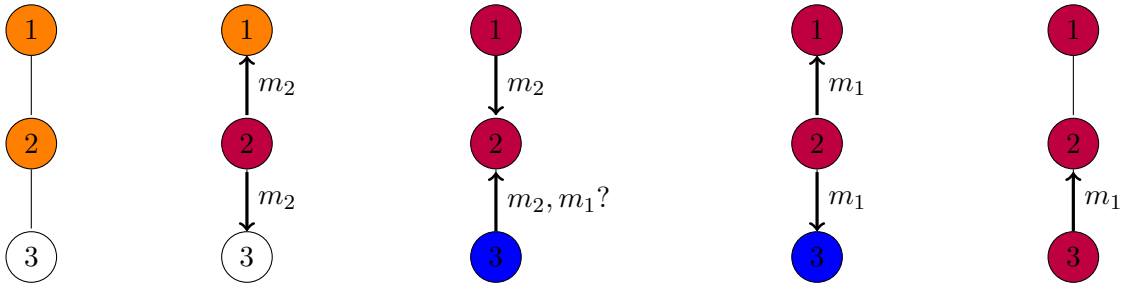


FIGURE 3.2 – Schéma du mécanisme de rattrapage du retard d'un nœud : le nœud 3 n'a, à l'origine (à gauche de la figure) pas reçu le message m_1 (il est alors représenté en blanc). Lors de la transmission du message m_2 (il devient bleu), il s'en rend compte et le demande à ses voisins. Le nœud 2 lui retransmet le message manqué (le nœud 3 devient alors violet).

attend trop longtemps, le nœud *rel* a plus de risques d'avoir quitté son voisinage et le message renvoyé n'a pas forcément d'intérêt pour les autres nœuds du réseau.

Algorithme 2 : **RDF - Détection d'un nœud en retard sur le nœud *id***

Initialisation :
  pour chaque $i \in \{1, \ldots, N\}$ **faire**
    pour chaque $j \in \{1, \ldots, N\}$ **faire**
      ▷ $M_{id}[l][m]$ *représente la connaissance par le nœud id du dernier message généré sur le nœud* l *que le nœud* m *a reçu.*
      $M_{id}[i][j] \leftarrow 0$
    fin pour
    $C_{id}[i] \leftarrow \{\}$
  fin pour
  $file\_envois_{id} \leftarrow []$

Nouvelle diffusion localement déclenchée :
  ▷ *La donnée payload doit être diffusé depuis le nœud id vers tous les autres nœuds*
  $nseq \leftarrow M_{id}[id][id]$
  $C_{id}[id][nseq] \leftarrow payload$
  append($file\_envois_{id}$,($id, nseq$))

Réception d'un message RDF :
  recevoir(*message*)
  $s \leftarrow message$.source
  $n \leftarrow message$.nseq
  $pld \leftarrow message$.payload
  ▷ *Identifiant du nœud émetteur du message*
  $rel \leftarrow message$.relay_node
  ▷ *Vecteur représentant l'état du nœud émetteur du message*
  $V_{rel} \leftarrow message$.relay_row
  ▷ *Transmission au nœud rel des messages qu'il a manqués et reçus par le nœud id*
  pour chaque q **tel que** $V_{rel}[q] < M_{id}[q][id]$ **faire** ▷ q *est un identifiant*

21 $n = V_{rel}[q] + 1$
22 **si** $n \in C_{id}[q]$ **alors**
▷ *Le message n généré par le nœud q doit être retransmis en priorité (ajouté en tête de liste file_envois$_{id}$)*
23 push(*file_envois*$_{id}$(q,n))
24 **fin si**
25 **fin pour**
▷ *Faire suivre le message si le message n'a pas encore été diffusé par le nœud id*
26 **si** $n > M_{id}[s][id]$ **et** $n \notin C_{id}[s]$ **alors**
27 $C_{id}[s][n] \leftarrow pld$
28 append(*file_envois*$_{id}$,(s, n))
29 **fin si**
▷ *Mise à jour de la matrice avec le numéro du dernier message (q) généré par le nœud s dont tous les prédécesseurs sont reçus*
30 $q \leftarrow n$
31 **tant que** $q + 1 \in C_{id}[s]$ **faire**
32 $q \leftarrow q + 1$
33 **fin tant que**
34 $M_{id}[s][id] \leftarrow q$

35 Expiration du timer d'envoi :
36 **si** *file_envois*$_{id} \neq \emptyset$ **alors**
37 $(s, n) \leftarrow$ pop(*file_envois*$_{id}$) ▷ *identifiant du message à envoyer*
38 $m \leftarrow$ nouveau_message()
39 m.source $\leftarrow s$
40 m.nseq $\leftarrow n$
41 m.payload = $C_{id}[s][n]$
▷ *identifiant id du nœud émettant le message*
42 m.relay_node $\leftarrow$ *id*
▷ *Vecteur représentant l'état du nœud id émettant le message*
43 m.relay_row $\leftarrow M_{id}[id]$
44 **envoyer(** m **)**
45 **fin si**
46 Réarmer le timer d'envoi

Il n'y a cependant aucune garantie que le nœud *id* sera encore voisin de *rel* lors du renvoi du message manqué (à cause de la mobilité des véhicules) ou que cet envoi réussira (à cause des interférences électromagnétiques). Il est donc possible que le renvoi de ce message ne fasse en réalité pas progresser la diffusion. Cependant, puisqu'il est toujours impossible dans un réseau véhiculaire ad hoc (VANET) de savoir si l'envoi d'un message sera utile ou non, cette absence de garantie n'est pas un argument suffisant pour l'empêcher de tenter de faire progresser une diffusion.

Il est possible d'envoyer au véhicule retardataire le premier message qu'il a manqué, mais également des messages plus récents. Comme le nœud retardataire a intérêt à ne pas recevoir plusieurs messages identiques, il est possible de choisir aléatoirement le message à renvoyer.

Différentes stratégies ont été explorées, et pourraient être mises en œuvre selon le contexte (densité du réseau, par exemple). Afin d'éviter une forte redondance (particulièrement si le réseau est dense), il est utile d'ajouter une dose d'aléatoire dans le choix du

message renvoyé [**?**]. Intuitivement, on peut simplement attribuer la même probabilité à tous les messages manqués par le retardataire qui peuvent être renvoyés. Une approche plus intermédiaire est d'attribuer les probabilités en fonction de l'ordre d'envoi des messages. Ainsi, il est possible d'associer la probabilité $\frac{1}{2}$ au premier message, $\frac{1}{4}$ au second et ainsi de suite. La probabilité résiduelle ($\frac{1}{2^n}$ où n est le nombre de messages) peut être ajoutée au premier message afin de le favoriser encore plus s'il y a peu de messages à rattraper.

3.4.3 Gestion de la mémoire

Lorsque tous les nœuds ont reçu une certaine donnée à diffuser, il devient inutile de la conserver en mémoire. De plus, en raison des ressources limitées dans les équipements embarqués dans les véhicules, il est important de supprimer les données inutiles de la mémoire. Cependant, il est difficile, pour un nœud, de savoir localement si tous les autres nœuds (y compris ceux qui ne font pas ou plus partie de ses voisins) ont reçu un message donné. Cette fonctionnalité nécessite de tenir à jour une représentation de l'état de tous les autres nœuds. C'est possible grâce à la matrice M_{id} représentant l'état de chaque nœud, vu par le nœud *id*. La détection d'un message bien diffusé dans un cas simple est présentée sur le schéma de la figure 3.3. Si la dynamique n'assure pas que tous les nœuds sont régulièrement voisins entre eux, il est important d'assurer une propagation de l'état des nœuds de proche en proche. Cette propagation peut s'effectuer en ajoutant la matrice du nœud émetteur lors de chaque transmission de message. Cependant, afin de limiter l'utilisation des ressources mémoire, on choisit de propager progressivement l'état des nœuds distants en ajoutant à chaque message une ligne (aléatoirement choisie) de la matrice de son émetteur concernant un autre nœud *sel*.

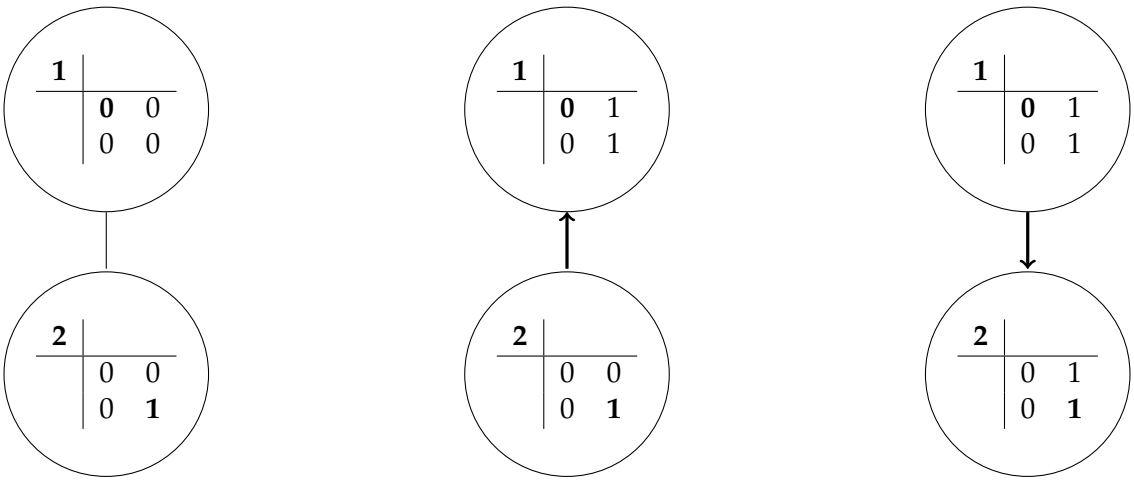


FIGURE 3.3 – Schéma représentant la détection de diffusion complète selon le temps qui s'écoule de gauche à droite. Chaque cercle représente un nœud et contient son identifiant ainsi que l'état de sa matrice. Le nœud 2 démarre une diffusion, et transmet le message au nœud 1. Ce dernier met sa matrice à jour et repère que tous les nœuds du réseau ont reçu le message. Il le supprime donc de sa mémoire. Le nœud 1 fait enfin suivre le message qui est reçu par le nœud 2 (à droite de la figure). Cette transmission permet au nœud 2 mettre à jour sa matrice et de remarquer que le message a été reçu par tout les nœuds du réseau. Il supprime alors également le message.

Lorsque la matrice (M_{id}) indique qu'un message (n généré sur le nœud i) a été reçu par tous les nœuds ($\forall s \in \{1,\ldots,N\} M_{id}[i][s] \geq n$), il est possible de le supprimer de la mémoire sans risque pour la fiabilité de la diffusion, comme indiqué dans la description complète de RDF à l'algorithme 3.

Algorithme 3 : **RDF sur le nœud *id***

```
1   Initialisation :
2     pour chaque i ∈ {1,...,N} faire
3       pour chaque j ∈ {1,...,N} faire
          ▷ M_id[l][m] représente la connaissance par le nœud id du dernier message généré sur
          le nœud l que le nœud m a reçu.
4         M_id[i][j] ← 0
5       fin pour
6       C_id[i] ← {}
7     fin pour
8     file_envois_id ← []
9   Nouvelle diffusion localement déclenchée :
      ▷ La donnée payload doit être diffusé depuis le nœud id vers tous les autres nœuds
10    nseq ← M_id[id][id]
11    C_id[id][nseq] ← payload
12    append(file_envois_id,(id, nseq))
13  Réception d'un message RDF :
14    recevoir( message )
15    s ← message.source
16    n ← message.nseq
17    pld ← message.payload
18    rel ← message.relay_node
19    V_rel ← message.relay_row
20    sel ← message.selected_node
21    V_sel ← message.selected_row
      ▷ Transmission au nœud rel des messages qu'il a manqués et reçus par le nœud id
22    pour chaque q tel que V_rel[q] < M_id[q][id] faire
        ▷ q est un identifiant
23      n = V_rel[q] + 1
24      si n ∈ C_id[q] alors
          ▷ Le message n généré par le nœud q doit être retransmis en priorité (ajouté en tête
          de liste file_envois_id)
25        push(file_envois_id(q,n))
26      fin si
27    fin pour
      ▷ Faire suivre le message si le message n'a pas encore été diffusé par le nœud id
28    si n > M_id[s][id] et n ∉ C_id[s] alors
        ▷ Sauvegarde en mémoire du message reçu
29      C_id[s][n] ← pld
        ▷ Transfert du message pour poursuivre la propagation
30      append(file_envois_id,(s, n))
```

```
31    fin si
      ▷ Mise à jour de la matrice avec le numéro du dernier message (q) généré par le nœud s
      dont tous les prédécesseurs sont reçus
32    q ← n
33    tant que q + 1 ∈ C_id[s] faire
34      q ← q + 1
35    fin tant que
36    M_id[s][id] ← q
      ▷ Suppression des messages correctement diffusés
37    pour chaque i ∈ {1, ..., N} faire
38      M_id[i][rel] ← max(M_id[i][rel], V_rel)
39      M_id[i][sel] ← max(M_id[i][sel], V_sel)
40    fin pour
41    pour chaque i ∈ {1, ..., N} faire
        ▷ mini est le numéro du plus récent message généré sur le nœud i dont le nœud id a
        connaissance de la bonne réception par tous les nœuds
42      mini = min(M_id[i])
43      pour chaque k dans C_id[i] faire
          ▷ Si le message k est plus ancien que mini, il a été reçu par tous et peut être détruit
44        si k ≤ mini alors
45          supprimer C_id[i][k]
46        fin si
47      fin pour
48    fin pour

49  Expiration du timer d'envoi :
50    si file_envois_id ≠ ∅ alors
51      (s, n) ← pop(file_envois_id)          ▷ identifiant du message à envoyer
52      m ← nouveau_message()
53      m.source ← s
54      m.nseq ← n
55      m.payload = C_id[s][n]
        ▷ identifiant id du nœud émettant le message
56      m.relay_node ← id
        ▷ Vecteur représentant l'état du nœud id émettant le message
57      m.relay_row ← M_id[id]
        ▷ identifiant id du nœud émettant le message
58      sel ← selection_aleatoire(1, ..., N)
59      m.selected_node ← sel
        ▷ Vecteur représentant l'état du nœud id émettant le message
60      m.selected_row ← M_id[sel]
61      envoyer( m )
62    fin si
63    Réarmer le timer d'envoi
```

L'algorithme RDF permet ainsi de diffuser une donnée, de vérifier à chaque émission que le nœud émetteur est à jour et de lui renvoyer les éventuels messages manqués, et enfin de ne pas conserver éternellement en mémoire de données inutiles, donc de limiter l'usage de cette ressource limitée. Dans les faits, on ajoute des émissions périodiques

si aucun message n'est en attente d'envoi pendant trop longtemps. Cela sert à déclencher le mécanisme de détection du retard chez les voisins du nœud émetteur après une inactivité trop longue.

Afin de valider cet algorithme, une évaluation de performance empirique est alors conduite.

3.5 Protocole pour l'évaluation de performances

Une étude expérimentale de l'algorithme est nécessaire, afin de valider le fonctionnement de l'algorithme de façon qualitative et d'évaluer les performances qu'il peut atteindre dans des conditions réalistes. Ces expériences nécessitent de déterminer les valeurs mesurées au cours de l'expériences, puis les métriques d'intérêt, avant d'établir un protocole expérimental.

3.5.1 Mesures expérimentales

Au cours d'une expérience, il est possible de collecter plusieurs données qui renseignent sur son comportement pendant l'exécution. Ces mesures sont utilisées pour établir des métriques permettant d'évaluer les performances atteintes lors de l'exécution.

L'algorithme de diffusion fiable a pour objectif la bonne diffusion du message à tous les nœuds du réseau. Son "bénéfice applicatif" peut donc être mesuré par le nombre de nœuds du réseau qui ont reçu le message à diffuser. Au cours d'une exécution, on peut alors le mesurer par la somme du nombre de messages reçus sur tous les nœuds. On définit donc la quantité $\textit{reçus}_n$ comme la quantité de messages (appartenant à des diffusions différentes) reçus par le nœud n au cours de l'exécution. La valeur $\textit{reçus}_n(t)$ représente le nombre de diffusions différentes ayant atteint le nœud n depuis le début de l'exécution et jusqu'à l'instant t (incluant les diffusions ayant démarré sur le nœud n).

Dans un réseau véhiculaire, les ressources réseau sont les plus critiques, à la fois parce que la bande passante est limitée mais aussi parce que le medium de communication est partagé avec tous les autres acteurs (véhicules, infrastructure, etc.). Ainsi, envoyer un message est coûteux pour tous les voisins de l'émetteur. En revanche, l'envoi d'un unique message peut donner lieu à la réception de multiples messages par différents nœuds. C'est pourquoi la mesure choisie pour représenter la consommation réseau de l'algorithme est la somme du nombre de messages émis $\textit{emis}_{\textit{source}}(t)$, par chaque nœud *source* depuis le début de l'exécution et jusqu'à l'instant t.

Le matériel embarqué constitutif des réseaux véhiculaires est généralement limité en mémoire. En effet, même si les véhicules peuvent embarquer suffisamment de ressources pour assurer le fonctionnement de systèmes complexes, leur mémoire reste finie et un algorithme ne peut consommer à chaque instant plus de mémoire que précédemment (fuite de mémoire). Pour ces raisons, il est important de vérifier que l'utilisation mémoire de l'algorithme reste stable. Le nombre de messages en cache $\textit{cache}_n(t)$ à l'instant t sur le nœud n est choisi pour mesurer l'utilisation mémoire.

Ces trois mesures rendent compte du bon fonctionnement de l'application, de l'utilisation du réseau et de la consommation mémoire. Elles sont utiles à la création de métriques, qui vont permettre l'évaluation expérimentale des performances de l'algorithme.

3.5.2 Métriques de performances

Afin d'évaluer correctement les performances atteintes par RDF, nous définissons les métriques d'intérêt. En effet, on peut toujours évaluer un algorithme sur toute une gamme de performances, dont il faut effectuer une interprétation. À l'aide des mesures définies précédemment, on définit les métriques utilisées pour analyser les performances atteintes par l'algorithme dans les expériences conduites.

L'algorithme RDF est conçu pour ne pas avoir de terminaison, il est toujours possible de relancer une nouvelle diffusion fiable. C'est donc l'évolution des indicateurs au cours du temps qui renseigne sur ses performances lors d'une exécution. En effet, tant que l'algorithme échoue à délivrer le message à tous les autres nœuds, il consomme des ressources afin d'y parvenir. Par ailleurs, indépendamment du temps d'exécution, chaque nouvelle diffusion enclenche la consommation de nouvelles ressources, par l'émission et la mémorisation de nouveaux messages. Cela montre l'importance, dans les métriques considérées, de normaliser les indicateurs en fonction du temps d'exécution et du nombre de diffusions ayant eu lieu.

Pour de multiples diffusions démarrant régulièrement, il est important que la consommation de ressources engendrée par les nouvelles diffusions soit compensée (en moyenne, tout du moins) par la libération de ressources lors de la résolution de diffusions précédentes.

La métrique réseau tiendra à la fois compte du nombre de messages envoyés, et du nombre de diffusions démarrées et de la période étudiée. Ainsi, la métrique réseau utilisée dans ce travail est, pour un réseau composé de N nœuds numérotés de 1 à N :

$$net(t) = \frac{1}{N} \times \sum_{i=1}^{N} \frac{emis_i(t)}{recus_i(t)} \tag{3.1}$$

Cette quantité est d'autant plus faible que le "bénéfice applicatif" est grand et d'autant plus importante que l'algorithme consomme de ressources. Pour un bon fonctionnement de l'algorithme, il est attendu que cette quantité se stabilise avec le temps, ce qui signifie que chaque nouvelle diffusion n'engendre pas (en moyenne) une consommation supérieure aux diffusions l'ayant précédée.

La valeur de cette métrique dépend de la topologie dynamique du réseau. Si elle augmentait en permanence au cours du temps, cela signifierait que l'ordre du nombre de messages émis est supérieur à l'ordre des messages reçus, et par conséquent, que l'exécution ne pourra pas se poursuivre sans atteindre les limites des capacités du protocole de communication utilisé.

$$O(\sum_{i=1}^{N} emis_i(t)) > O(\sum_{i=1}^{N} reçus_i(t))$$

La métrique de l'utilisation des ressources mémoire doit permettre de s'assurer simplement que la quantité moyenne de mémoire utilisée au cours du temps $\sum_{i=1}^{N} cache_i(t)$ reste stable pendant l'exécution. Si cela n'était pas le cas, la mémoire libérée par la résolution de diffusions anciennes ne compenserait pas la mémoire consommée par les nouvelles diffusions. Une telle situation assure que l'algorithme finira par consommer toutes les ressources mémoire dont il dispose et échouera. Plus concrètement, une augmentation perpétuelle de la quantité $\sum_{i=1}^{N} cache_i(t)$ assure que le nombre de diffusions en cours (démarrées mais non terminées) augmente au cours du temps et donc que la

Performance	Métrique
"Bénéfice applicatif"	$mem(t)$
Réseau	$net(t)$
Mémoire	$mem(t)$

TABLE 3.1 – Métriques d'étude des performances de l'algorithme de diffusion fiable RDF.

durée moyenne des diffusions augmente au cours du temps. L'utilisation d'une quantité non stable (bornée) au cours du temps de mémoire assure donc un échec algorithmique. Pour cette raison, la métrique utilisée pour rendre compte des performances concernant l'utilisation mémoire est :

$$mem(t) = \frac{1}{N} \times \sum_{i=1}^{N} \frac{cache_i(t)}{reçus_i(t)} \tag{3.2}$$

Pour que RDF fonctionne correctement, il est attendu que cette métrique tende vers 0 (après une période de transition) ce qui signifierait que la consommation mémoire reste, en moyenne stable. Si elle tend vers une autre valeur ou n'admet pas de limite, l'algorithme utilise une quantité non stable de mémoire et finit donc par échouer à terminer autant de diffusion qu'il en démarre, ce qui signifie que la durée de diffusion ne pourra être bornée.

Ces métriques résumées en table 3.1 permettent d'étudier le comportement de l'algorithme RDF à la fois en termes de ressources consommées et en terme de "bénéfice applicatif".

3.5.3 Expériences réalisées

L'évaluation de performances de l'algorithme nécessite un protocole expérimental. Ce dernier vise à permettre de valider le fonctionnement de l'algorithme en conditions réelles, ce qui nécessite tout d'abord une implémentation de l'algorithme, puis son intégration au sein d'une expérience de Preuve de Concept (PoC). Le protocole expérimental a également pour objectif de définir des expériences permettant de quantifier les résultats de l'exécution de l'algorithme, via les métriques précédemment décrites.

Les expériences destinées à l'évaluation de l'algorithme réparti nécessitent une implémentation de l'algorithme RDF. L'intergiciel (middleware) Airplug, décrit au chapitre 2 est utilisé pour cette étude. Il permet en effet d'effectuer à la fois des expériences en direct et des études de scénarios émulés en laboratoire en utilisant la même implémentation [**?**].

L'application RDF est en charge de disséminer sur tous les nœuds impliqués des messages générés localement par une autre application. Elle est implémentée comme un programme autonome (s'exécutant dans son propre processus Linux) dupliqué sur les nœuds du réseau. Son implémentation est réalisée en Python, en utilisant des listes pour stocker les états des nœuds (la matrice est une liste de N listes de N entiers sur chaque nœud) et un dictionnaire pour stocker les messages. Les messages de l'application RDF transmis d'un nœud à l'autre sont composés de 6 champs :

— l'identifiant du nœud générant la charge utile ;

— le numéro de séquence du message ;

— l'état et l'identifiant du nœud émetteur ;

— l'état et l'identifiant d'un autre nœud (sélectionné pour propager la connaissance des caches réseau) ;

— l'application de destination, intéressée par la charge utile ;

— la charge utile, c'est-à-dire la donnée générée à diffuser.

L'implémentation de l'application RDF est constituée d'un programme Python de 400 lignes environ, utilisant les fonctionnalités des bibliothèques Airplug facilitant le développement d'applications réparties. Elle inclut une interface graphique laissant paraître facilement les champs du dernier message reçu, mais aussi la matrice interne de chaque nœud, et le nombre de messages en mémoire, comme le montre la figure 3.4.

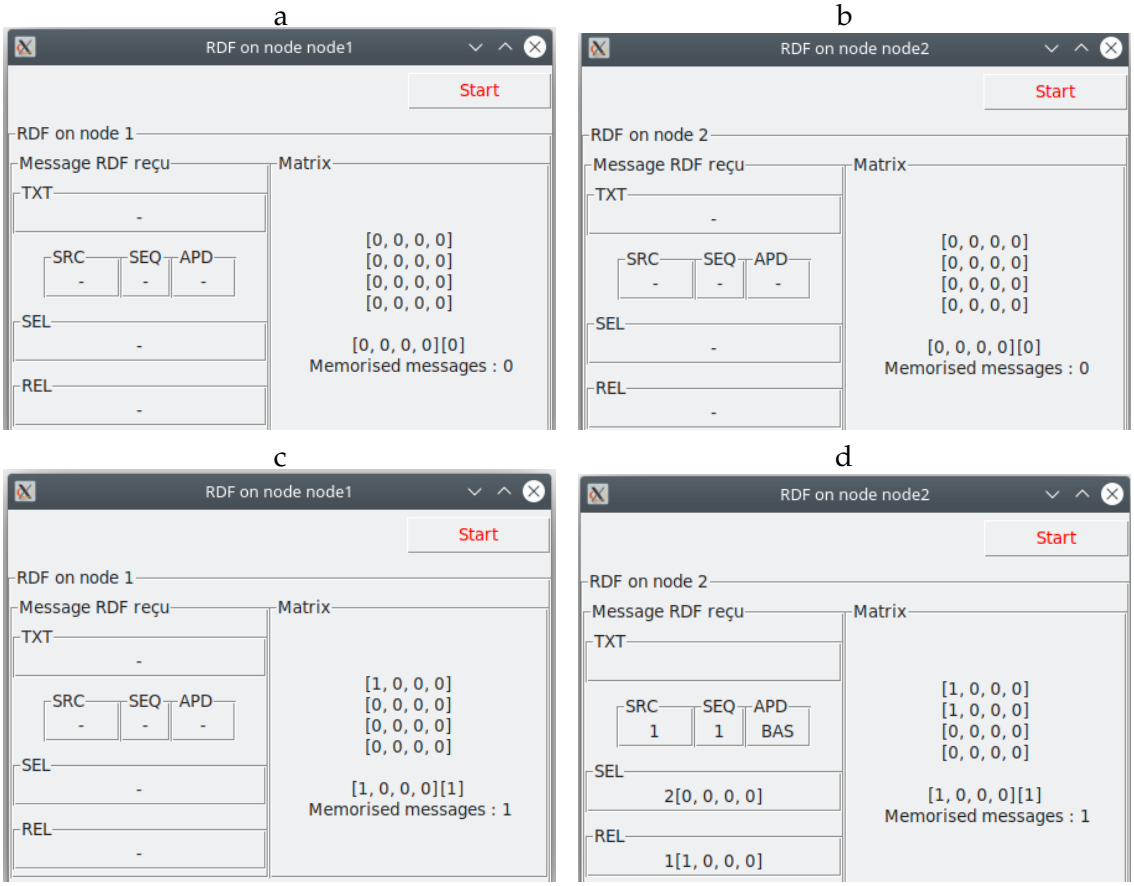


FIGURE 3.4 – Captures d'écran de l'application RDF sur les nœuds 1 et 2 d'un scénario concernant 4 nœuds. La figure a représente l'application RDF sur le nœud 1 avant la génération de tout message à diffuser tandis que la figure b représente RDF sur le nœud 2 dans les mêmes circonstances. La figure c représente l'application RDF sur le nœud 1 après qu'un message a été généré (à la demande d'une application locale nommée BAS) et envoyé à ses voisins et la figure d représente le nœud 2, voisin de nœud 1, qui a reçu le message.

Les communications entre l'application RDF et les autres applications sont assurées par des messages locaux plus simples que les messages entre instances différentes de RDF (il y a moins de données à transmettre : uniquement la charge utile et le nom de l'application). Cette implémentation peut être utilisée directement grâce aux fonction-

nalités d'Airplug, à la fois pour des expériences réelles que pour des expériences en émulation.

Une validation de l'application en conditions réelles est tout d'abord conduite afin de s'assurer de son fonctionnement de manière qualitative. Cette expériennce n'a qu'une valeur de démonstration conceptuelle mais permet de s'assurer que l'application remplit bien ses objectifs.

Cette expérience est réalisée sur 8 petits robots communiquant par une liaison WiFi en mode ad hoc. Ces robots (tels que celui présenté en figure 3.5) sont pilotés par une application qui leur permet de communiquer à leurs voisins des instructions et leurs données de positions. Cette application sera utilisée dans deux situations différentes :

— seule, pour s'assurer qu'elle n'assure pas déjà une diffusion fiable des données dans le scénario étudié, certains robots devraient alors manquer des instructions,

— en communiquant via l'application RDF, pour s'assurer que tous les nœuds reçoivent toutes les instructions données par un robot leader.

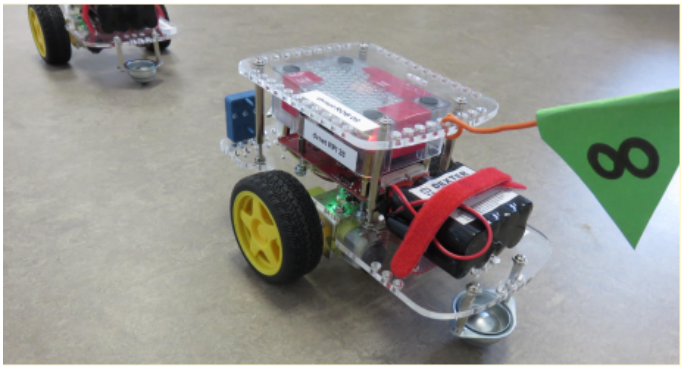


FIGURE 3.5 – Photographie d'un robot utilisé dans l'expérience de preuve de concept de l'algorithme de diffusion fiable RDF.

Afin d'évaluer plus quantativement les performances permises par l'application RDF, des tests en émulation sont conduits, grâce à l'émulateur Airplug décrit au chapitre 2, qui relie les différentes applications réparties et les différents nœuds par des pipes. Les trajectoires des nœuds mobiles sont gérées directement par l'émulateur capable d'intégrer par exemple des traces GPS issues de tests routiers.

Pour réaliser une diffusion, la pire topologie possible est la chaîne topologique, puisque la hauteur de l'arbre couvrant dans un tel réseau est la longueur de la chaîne, soit le nombre de nœuds (à un près), comme montré sur la figure 3.6.

De plus, dans un contexte véhiculaire, ce type de topologie correspond à un convoi de véhicules largement espacés, sur une autoroute, par exemple. Les expérimentations se focaliseront donc sur cette topologie avec 4 nœuds, dans laquelle chaque nœud déclenchera périodiquement une nouvelle diffusion (à une fréquence de 1 Hz). L'application RDF est paramétrée pour permettre d'envoyer au maximum 10 messages par seconde.

Dans le cas d'une chaîne topologique, chaque nœud retardataire n'est entouré au maximum que d'un nœud à jour. Pour cette raison, il est toujours plus pertinent de renvoyer le premier message manqué. Afin de mettre en valeur les différences entre les deux stratégies, un scénario comprenant 2 convois de 4 véhicules est également testé. L'émulateur est successivement paramétré avec plusieurs valeurs de taux de pertes (probabilité qu'un message émis ne soit pas reçu).

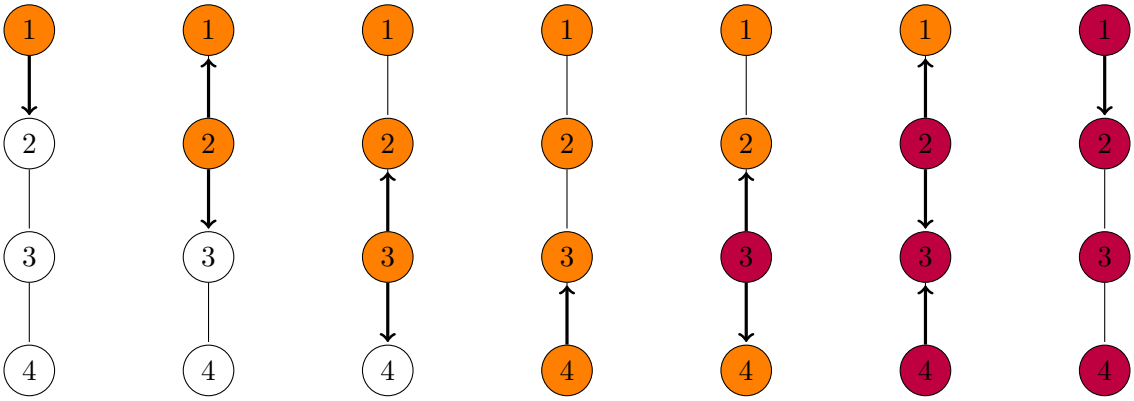


FIGURE 3.6 – Exemple de diffusion en topologie chaîne de longueur 4. La topologie est inconnue par les nœuds du réseau et le temps s'écoule de gauche à droite. La première diffusion part du bout de la chaîne (nœud 1) à gauche de l'image, et se propage par l'émission de 4 messages. La seconde part du milieu de la chaîne (nœud 3), et nécessite également l'émission de 4 messages pour se propager dans toute la chaîne.

Les deux topologies utilisées lors des tests sont représentées sur la figure 3.7.

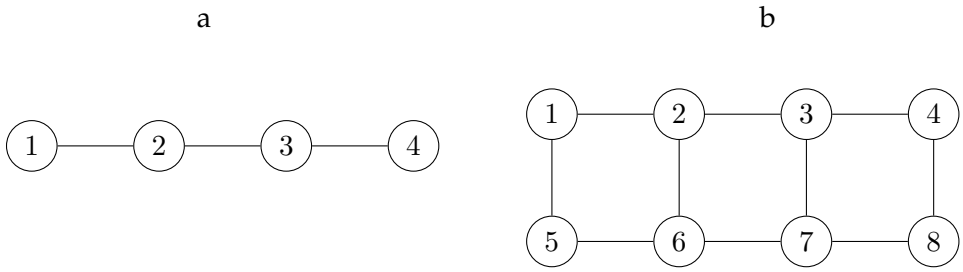


FIGURE 3.7 – Topologie des tests conduits en émulation. Sur la figure a, la topologie d'un convoi correspondant au pire cas, et sur la figure b, celle correspondant au scénario à deux convois

3.6 Résultats des expériences de validation

Les résultats et interprétations des expériences proposées à la section précédente sont présentés ci-après.

3.6.1 Démonstration conceptuelle (PoC)

Au cours de la démonstration conceptuelle, les robots se transmettent des instructions et leurs données grâce à une application de pilotage.

Lorsque l'application de pilotage est directement utilisée pour envoyer des instructions aux robots, chacun étant à portée de tous les autres, on constate que certaines instructions ne sont malgré tout pas reçues par certains robots, qui peuvent reprendre une marche normale après avoir ignoré une ou plusieurs commandes. Cela semble dû à des pertes de messages, liées à de possibles collisions ou à l'environnement électromagnétique.

Lorsque l'application RDF est utilisée pour faire transiter les messages d'un robot vers les autres, ces comportements disparaissent et les robots réalisent tous les mêmes instructions, même si certains prennent parfois un peu de retard.

Cette expérience permet de valider le bon fonctionnement de l'application dans une situation réelle simple (le réseau est complet mais non fiable).

3.6.2 Performance réseau en pire cas

Dans le cadre d'un scénario où les nœuds forment un convoi ayant la topologie d'une chaîne, la performance réseau ($net(t)$) est représentée sur la figure 3.8 en fonction du temps, pour plusieurs valeurs de fiabilité de canal (la fiabilité du canal est la valeur complémentaire du taux de perte). Le choix du message renvoyé se tourne systématiquement vers le premier message manqué.

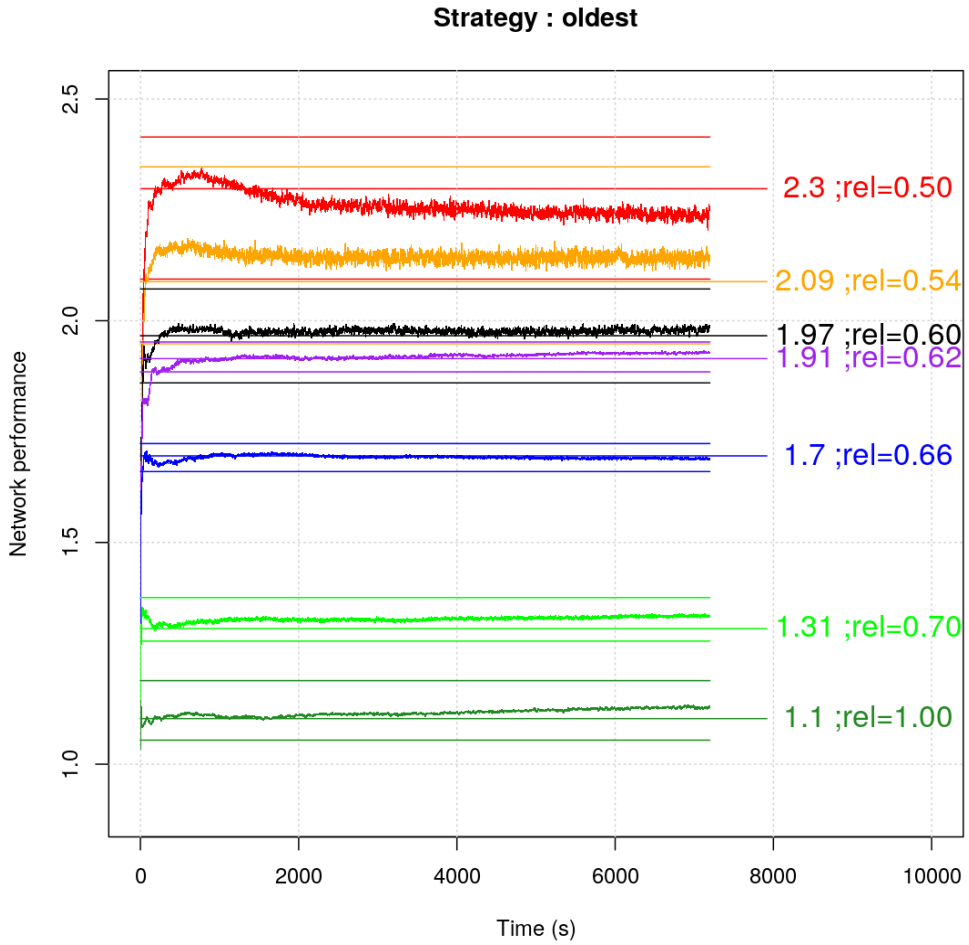


FIGURE 3.8 – Performances réseau de l'algorithme RDF (sous forme de moyenne mobile au cours du temps) dans un scénario de convoi de 4 véhicules connectés en chaîne topologique avec renvoi du premier message manqué. Pour chaque série de valeurs (correspondant à une valeur de fiabilité du canal de communication notée *rel*), la médiane et les quartiles sont indiqués

Les résultats présentent une importante variabilité en fonction du temps, malgré une

tendance nette. Pour les interpréter plus facilement, une moyenne mobile est réalisée sur les valeurs, mais la dispersion reste indiquée à travers la médiane les quartiles. On peut y observer que la performance réseau se stabilise systématiquement assez rapidement autour d'une valeur (qui augmente avec le taux de perte), sans jamais surcharger le réseau, même si un message sur deux est perdu (pour des taux de pertes encore plus grands, le résultat est similaire).

Dans un environnement sans pertes de messages, cette valeur est de $1,1$, là où l'algorithme PIF [**?**], par exemple, qui n'assure pas la fiabilité de la diffusion et est plutôt pensé pour une topologie fixe (quoi qu'inconnue) donnerait un résultat de $1,5$. En effet, deux messages circuleraient sur chacun des 3 liens : la diffusion puis l'accusé de réception. L'algorithme RDF, quant à lui, a une performance réseau meilleure (environ $1,3$) même lorsque le taux de perte est de 30 % (fiabilité $rel = 0,70$). Comme ces différents algorithmes n'ont pas la même taille de message, une performance réseau plus grande n'assure pas forcément un usage plus grand de bande passante, mais dès lors que la transmission est sans fil et qu'elle implique un protocole qui va alourdir le message, la différence de taille des message peut être moins visible dans la probabilité d'interférer avec les messages des autres.

3.6.3 Performance mémoire en pire cas

Dans le cadre du scénario où les véhicules forment un convoi ayant la topologie d'une chaîne, la performance mémoire ($mem(t)$) est représentée en fonction du temps sur la figure 3.9 lorsque la stratégie de renvoi consiste à renvoyer le plus ancien message manqué.

On peut remarquer qu'elle tend rapidement vers 0 dès que le taux de perte est inférieur à 40 % ($rel = 0,60$), ce qui signifie que la fiabilité est assurée par l'algorithme malgré un taux de pertes élevé et une topologie très peu propice à la diffusion (chaîne topologique).

La stratégie affectant à chaque message une probabilité croissant exponentiellement avec son ancienneté donne des résultats similaires dans ce scénario, comme en témoigne la figure 3.10. C'est lié au fait que la topologie interdit toute redondance des messages puisqu'un nœud retardataire n'a, au plus qu'un unique voisin à jour.

Les résultats présentent une importante variabilité en fonction du temps, malgré une tendance nette. Pour les interpréter plus facilement, une moyenne mobile est réalisée sur les valeurs, mais la dispersion reste indiquée à travers la médiane les quartiles.

3.6.4 Comparaison des stratégies de renvoi de messages

Dans le contexte d'un convoi double, la connectivité est largement augmentée, ce qui permet à l'algorithme d'assurer la fiabilité de la diffusion malgré un environnement électromagnétique bien plus défavorable comme le montre la figure 3.11. Ainsi, même avec la perte de 4 messages sur 5 ($rel = 0,20$), la fiabilité est assurée avec les deux stratégies. On peut tout de même remarquer que la stratégie disposant d'une dose d'aléatoire (*exponent*) permet à la taille de cache de se stabiliser (arrêter d'augmenter) plus rapidement, ce qui signifie que la consommation de ressources mémoires sera mécaniquement plus faible avec cette stratégie dans ces circonstances.

Les vérifications expérimentales montrent que l'algorithme RDF assure bien une diffusion fiable, même dans des circonstances très défavorables à la résolution de ce pro-

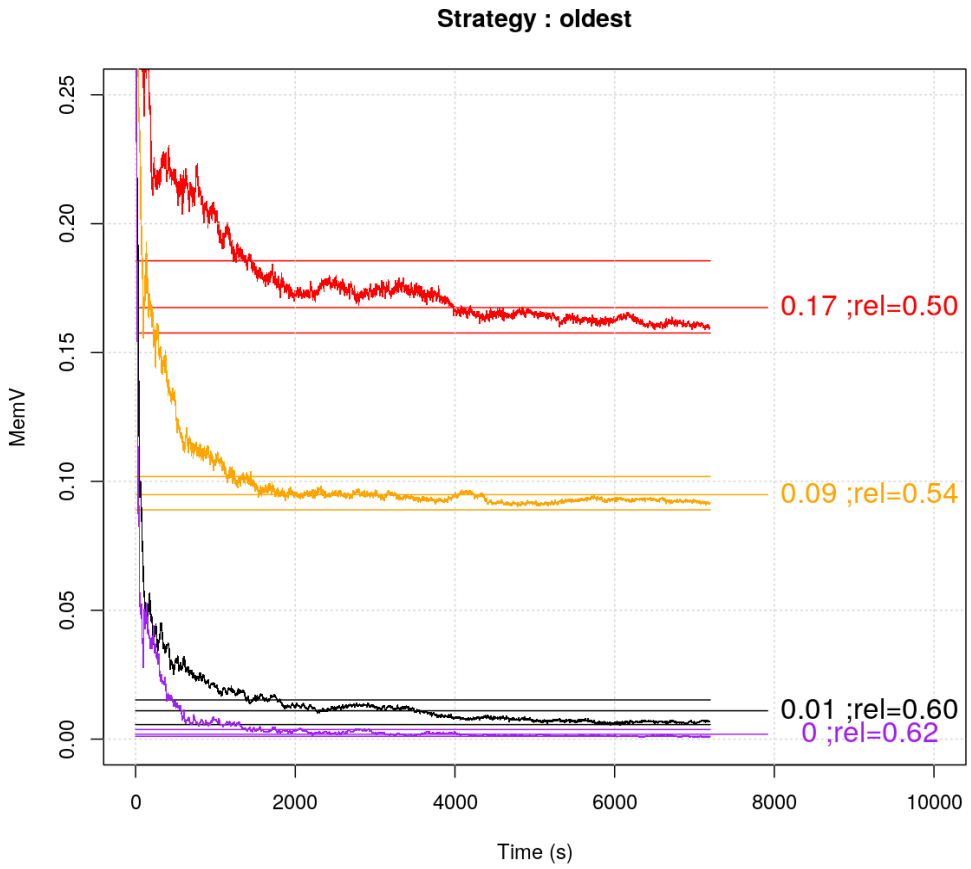


FIGURE 3.9 – Performances mémoire de l'algorithme RDF (sous forme de moyenne mobile au cours du temps) dans un scénario de convoi de 4 véhicules connectés en chaîne topologique avec renvoi du premier message manqué. Pour chaque série de valeurs (correspondant à une valeur de fiabilité du canal de communication), la médiane et les quartiles sont indiqués.

blème (faible connectivité et faible fiabilité du canal de communication). Il évite de surcharger le réseau (comparable à des algorithmes de diffusion non fiable) et assure l'utilisation d'une quantité finie de mémoire malgré des pertes importantes.

Si les conditions se dégradent trop, l'algorithme consommera toute sa mémoire (ressource locale) mais ne surchargera pas plus le réseau (ressource partagée), évitant ainsi à un nœud ayant de mauvaises conditions réseau de pénaliser les autres. Dans de telles circonstances, un algorithme n'assurant pas la fiabilité de la diffusion reste utilisable.

3.7 Conclusion

Dans un réseau véhiculaire, les problèmes les plus simples deviennent facilement complexes à résoudre. C'est par exemple le cas de la diffusion, puisqu'elle doit se faire dans une topologie inconnue, qui nécessite l'envoi de messages supplémentaires.

Un algorithme de diffusion fiable nommé RDF a été conçu, il est basé sur des émissions périodiques permettant de repérer si un nœud a manqué un message. Tous les nœuds du réseau gardent les messages en cours de diffusion en mémoire et suppriment

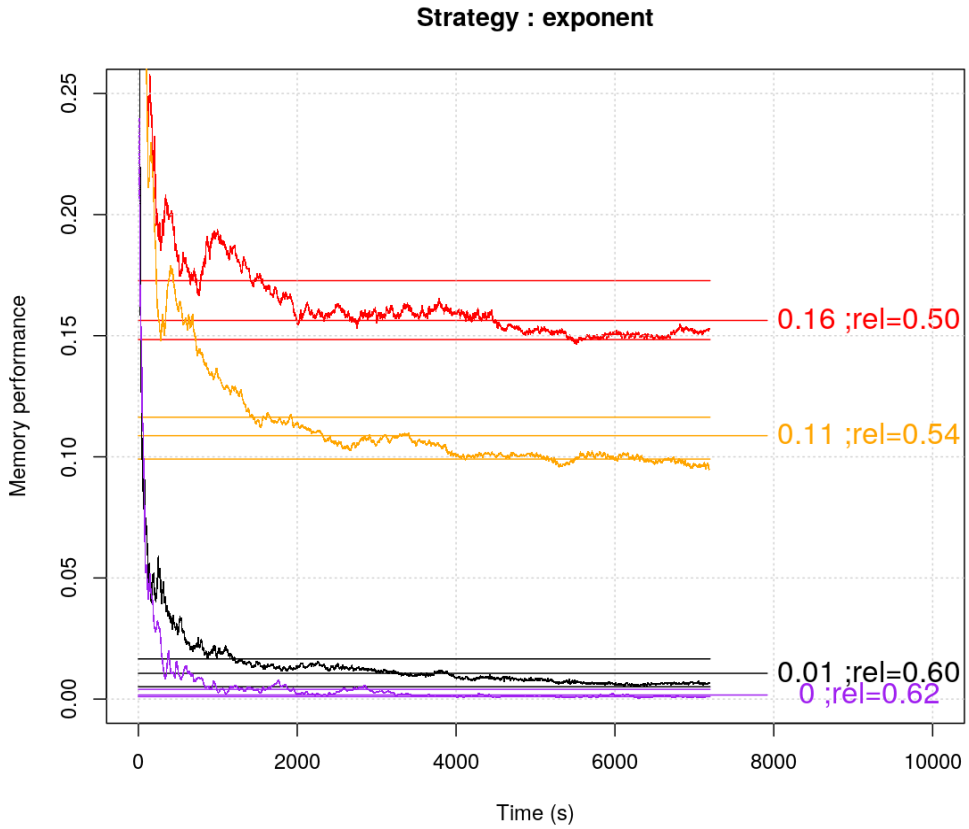


FIGURE 3.10 – Performances mémoire de l'algorithme RDF (sous forme de moyenne mobile au cours du temps) dans un scénario de convoi de 4 véhicules connectés en chaîne topologique avec renvoi aléatoire (probabilité croissant exponentiellement avec l'ancienneté) d'un message manqué. Pour chaque série de valeurs (correspondant à une valeur de fiabilité du canal de communication), la médiane et les quartiles sont indiqués.

ceux dont la diffusion est complète.

L'étude de performance de l'algorithme RDF montre que l'algorithme est capable d'assurer la diffusion fiable malgré des conditions très difficiles, avec une topologie réseau très défavorable (chaîne) et d'importantes pertes de messages (40 %). Les ressources utilisées par RDF s'avèrent limitées, tant du point de vue réseau que mémoire. En effet, la taille de la mémoire se stabilise rapidement au début de l'exécution tandis que la consommation réseau est comparable à des protocoles de diffusion sans mécanismes de fiabilité.

Les résultats de l'étude de performances sont cependant difficiles à extrapoler à une situation véhiculaire réelle différente, où l'espace entre les véhicules et la portée du protocole de communication pourraient varier, par exemple.

Ce chapitre a ainsi montré comment les performances d'une application de coopération véhiculaire sont tributaires des conditions d'exécution, et que les études expérimentales de performances ne fournissent pas une validation suffisamment solide pour justifier un déploiement coûteux. Une validation plus solide pourrait alors faire appel à des modélisations théoriques du réseau dans lequel s'exécutera l'application déployée.

Ces modélisations sont donc étudiées au chapitre 4.

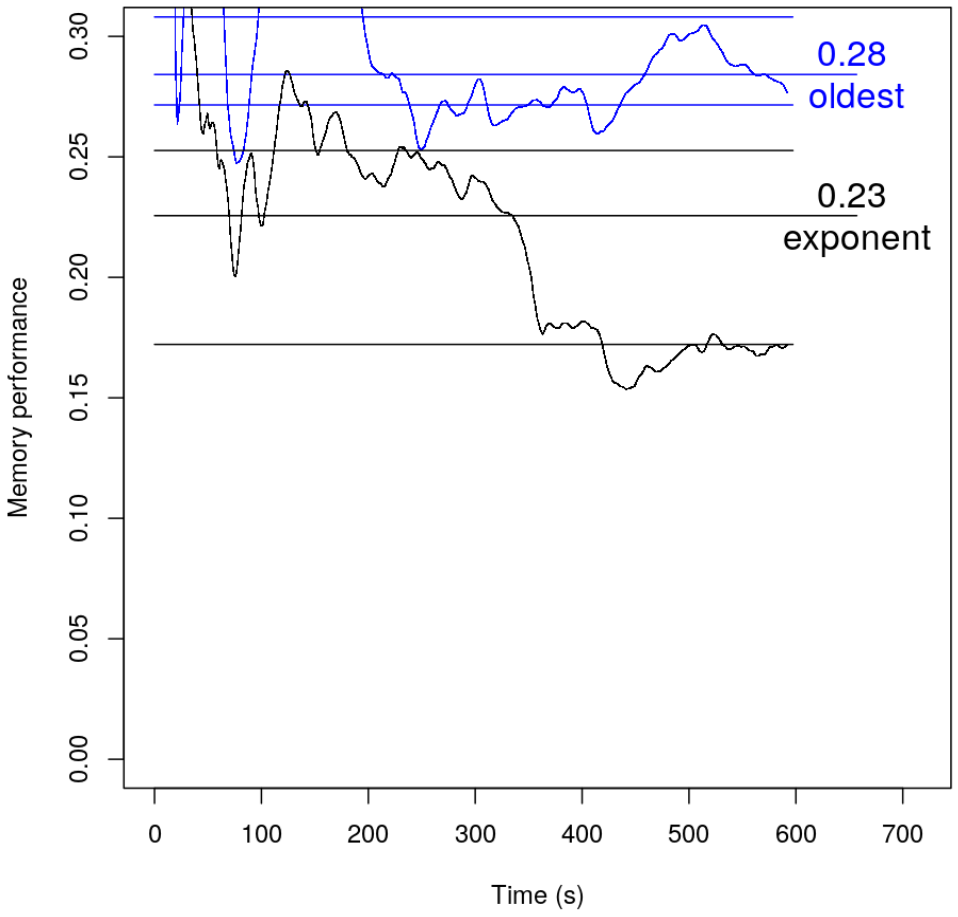


FIGURE 3.11 – Performances mémoire de l'algorithme RDF (sous forme de moyenne mobile au cours du temps) dans un scénario de de double convoi connecté en grille topologique de 8 véhicules et un taux de perte de messages de 80 %. Pour chaque série de valeurs (correspondant à une stratégie différente de renvoi de message), la médiane et les quartiles sont indiqués

Chapitre 4

Modélisation de la topologie d'un réseau dynamique de véhicules

Sommaire

4.1 Introduction

La coopération dans les VANET se heurte à la dynamique des véhicules. Il peut alors être nécessaire de modéliser la topologie dynamique d'un réseau véhiculaire afin de pouvoir étudier la coopération qui s'y effectue. Cette modélisation doit tenir compte des caractéristiques de ces réseaux tout en permettant de mettre en lumière des propriétés intéressantes pour leur étude.

La littérature scientifique s'est penchée sur les manières de modéliser la topologie des réseaux depuis des décennies et ces modèles reposent sur des graphes lorsque l'étude de la topologie d'un réseau statique se situe à haut niveau d'abstraction, comme par exemple [**?**] qui propose un algorithme de consensus, ou [**?**], qui propose un algorithme de diffusion. Dans les deux cas, l'étude de l'algorithme se fait en représentant la topologie réseau par un graphe dont les propriétés (connexité, présence de cycles) permettent la démonstration de l'algorithme. Nous nous intéressons tout d'abord aux premiers efforts pour adapter les modèles à base de graphes à la présence de changements de topologie,

avant d'étudier le modèle des p-graphes dynamiques, qui nous semble pertinent pour l'étude des réseaux de véhicules.

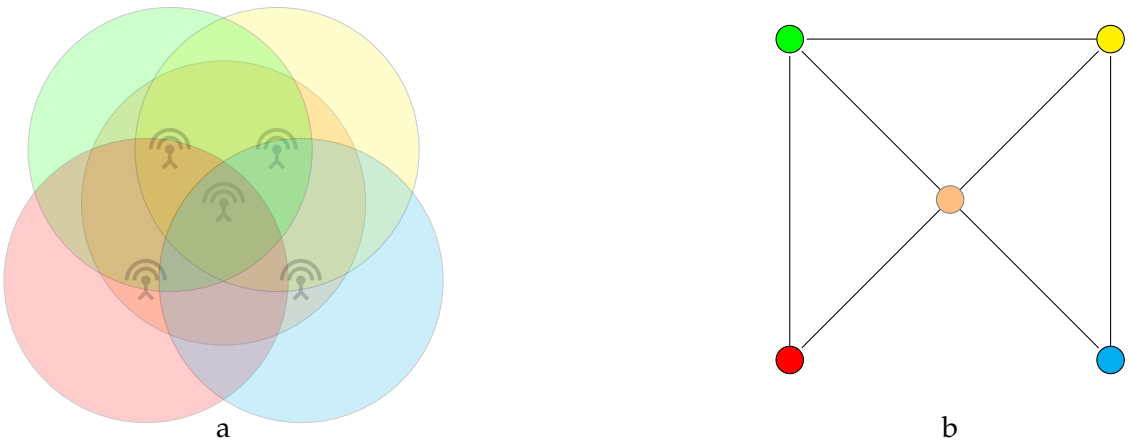
4.2 Modèle des graphes évolutifs

Il est possible de modéliser la topologie d'un réseau statique à l'aide d'un graphe, dans lequel les nœuds représentent les machines du réseau et les arêtes les liens de communication (habituellement non orientés) disponibles entre ces machines. En se basant sur cette modélisation, on peut construire un modèle adapté aux réseaux dont la topologie est dynamique comme les réseaux véhiculaires.

4.2.1 Graphes statiques en topologie fixe

Lorsque le réseau est filaire, les arêtes correspondent à des câbles de liaison du réseau. La géométrie du réseau matériel est ainsi semblable au graphe. En dehors des paramètres de la topologie réseau, le modèle ne tient généralement pas compte des paramètres réseau (bande passante, latence, détails protocolaires, gestion des collisions). Ainsi, l'étude du problème de l'arbre couvrant (*Spanning Tree*) dans un graphe dans [**?**] a notamment permis le développement du protocole réseau STP dans [**?**], avec ces hypothèses. Puisque la gestion des collision peut se faire à bas niveau (couche physique ou liaison) et assurer la fiabilité des liens pour les couches réseau supérieures, la modélisation à haut niveau du réseau peut faire l'hypothèse de liens parfaits.

Cependant, lorsque les liens sont sans fil, modéliser un réseau sous forme de graphes est une tâche plus complexe. En effet, il devient nécessaire de tenir compte de certains paramètres techniques du protocole de communication. On doit prendre en compte notamment la portée du medium de communication en plus de la position géographique des nœuds du réseau. On ajoute alors un lien lorsque deux nœuds sont suffisamment proches pour communiquer (la distance les séparant est inférieure à la portée de leur système de communication). On construit alors un graphe de disques unitaires, tels que présentés dans [**?**], selon la méthode décrite sur le schéma à la figure 4.1.



a b

FIGURE 4.1 – Schéma de définition de la topologie sans fil. Sur la figure a, une représentation du matériel en place avec la portée du protocole de communication utilisé, et sur la figure b, le graphe représentant la topologie associée.

Les spécificités des communications radio sans fil doivent également être prises en compte. En particulier la qualité des liens qui ne peuvent plus être supposés fiables. En effet, les interférences radio et les collisions d'émissions peuvent imposer des délais d'attente assez longs aux nœuds lorsque l'environnement radio est défavorable. D'autre part, s'il est possible, dans un réseau filaire, de définir le destinataire d'une émission en spécifiant le port d'émission, cela n'est pas le cas dans un réseau sans fil. Cela signifie par exemple qu'une émission unicast est techniquement identique à un broadcast, dans lequel le destinataire est précisé, et qui sera donc ignoré par les autres nœuds.

Dans un réseau sans fil, les collisions sont particulièrement problématiques. Dans un réseau sans infrastructure prédéfinie, il est donc impossible de repérer les collisions et de fiabiliser les communications en reproduisant les émissions perdues. Pour tenir compte de ces particularités, on peut étudier le réseau à travers le prisme de la tolérance aux fautes, en considérant que les liens peuvent être fautifs, comme par exemple dans [**?**, **?**].

4.2.2 Séquences de graphes

Dans un environnement routier, les véhicules sont habituellement en mouvement, et changent donc fréquemment de position. Ces changements de position peuvent alors provoquer des changements topologiques, comme le montre la figure 4.2, lorsque des véhicules s'éloignent.

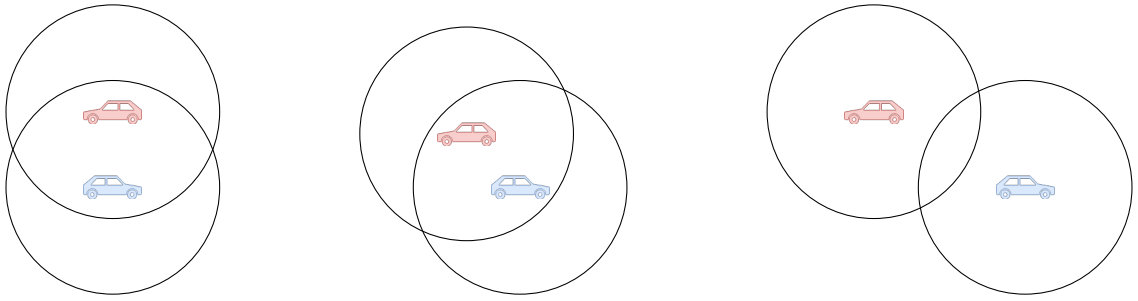


FIGURE 4.2 – Schéma de l'éloignement de deux véhicules avec un système de communication de portée fixe.

Si la modélisation par graphe est insuffisante pour rendre compte des changements topologiques apparaissant dans un réseau dynamique, on peut cependant se baser dessus pour obtenir un modèle intuitif d'un réseau dynamique. Ainsi, il est possible de modéliser un réseau dynamique à l'aide d'une séquence de graphes [**?**]. Chaque graphe représente une des topologies adoptées par le réseau au cours du temps, comme montré à la figure 4.3. Il peut être étiqueté avec deux dates (la date de début et la date de fin). Chaque changement topologique donne lieu à l'ajout d'un nouveau graphe dans la séquence. Dans ces circonstances, la date de début d'un graphe est donc forcément identique à la date de fin du graphe le précédant dans la séquence. Selon la période à laquelle l'étude s'intéresse, il devient possible d'étudier la topologie réseau de cette période à l'aide du modèle des graphes (statiques).

La séquence de graphes obtenue est une représentation sans pertes d'information de la topologie dynamique, qui sera appelée, dans la suite de ce travail, *observation de la dynamique du réseau*.

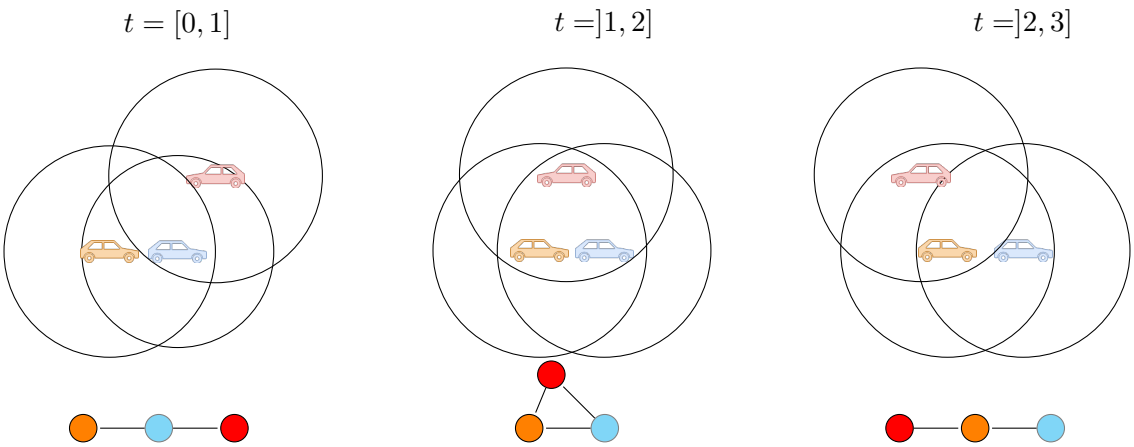


FIGURE 4.3 – Exemple de topologie dynamique d'un réseau véhiculaire représentée par une séquence de graphes.

Si la période d'intérêt traverse un changement topologique, il est impossible de l'étudier directement. On peut parfois couper l'étude en deux parties : une pour chaque topologie impliquée, mais c'est parfois impossible en fonction de la nature de l'étude à mener. En effet, si l'étude se concentre sur des actions atomiques dont la durée est supérieure à celle d'une des topologies impliquées dans la période d'intérêt, il devient impossible de mener cette étude sur la période d'intérêt.

Lorsque les changements topologiques sont trop fréquents, ces exceptions deviennent fréquentes et ne peuvent plus être ignorées. Le modèle doit alors disposer de propriétés permettant l'étude de de la topologie des réseaux fortement dynamiques à travers des changements topologiques.

Une adaptation de ce modèle est proposé par [**?**, **?**] sous le nom de *evolving graph* après l'apparition de travaux [**?**] montrant la nécessité de prendre en compte les propriétés des changements topologiques à l'intérieur du modèle. Ce modèle, basé sur une séquence de graphes, ajoute des propriétés permettant de prendre en compte les changements topologiques.

Le principal apport est la définition d'un chemin temporel (nommé *journey*), qui est constitué d'une séquence ordonnée d'arêtes du graphe dynamique au sein de laquelle chaque arête est associée à une date antérieure à la suivante. Chaque arête doit exister à la date à laquelle elle est associer. Un chemin temporel a alors un début et une fin (parfois appelés départ et arrivée), et est donc orienté, comme le montre la figure 4.4, même si les arêtes qui le composent ne le sont pas. En effet, dans cette situation, un chemin temporel permet de communiquer du nœud 1 au nœud 6 via le nœud 3 mais aucun ne permet d'aller du nœud 6 au nœud 1. Les propriétés de la topologie dynamique sont différentes des propriétés des graphes de la séquence. Par exemple, deux nœuds situés dans des composantes connexes différentes sur chaque graphe de la séquence peuvent cependant être reliés par un chemin temporel.

Il est possible d'étudier de manière indépendante la longueur topologique (nombre de sauts) du chemin temporel et sa durée (différence entre date de la dernière arête et de la première). Ces notions sont utilisées dans [**?**] pour la conception d'un algorithme de routage adapté à des changements topologiques prédictibles.

Le concept de chemins temporels et leurs propriétés permettent d'envisager de trans-

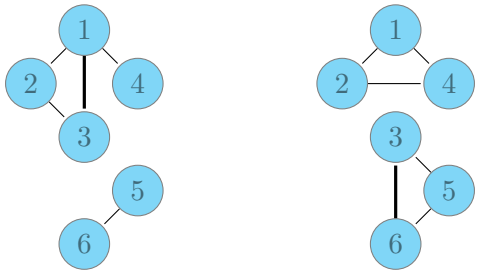


FIGURE 4.4 – Exemple de topologie dynamique. Les arêtes en gras peuvent faire partie d'un chemin temporel (*journey*) reliant le nœud 1 au nœud 6 via le nœud 3. Les nœuds 1 et 6 ne font jamais partie de la même composante connexe et aucun chemin temporel ne permet de relier le nœud 6 au nœud 1.

former une partie des propriétés des graphes statiques à ce modèle dynamique.

4.3 Modèle des graphes dynamiques (TVG)

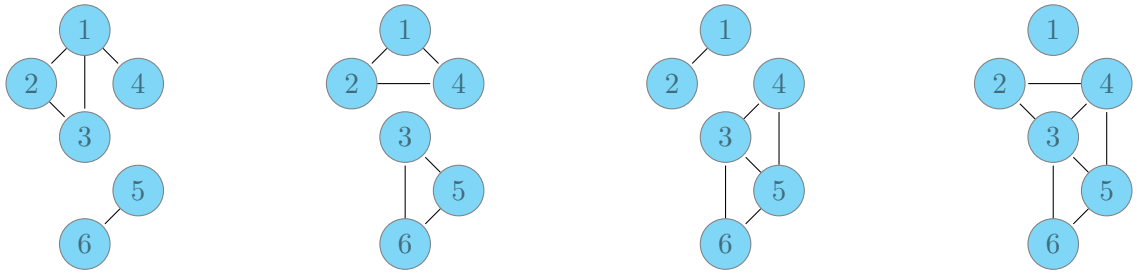
Une formalisation des graphes dynamiques nommée *TVG* pour *Time-Varying Graphs* est proposé par [**?**], se basant sur les graphes évolutifs. Ce modèle s'abstrait de la notion de séquence de graphes, et propose même des représentations de la topologie dynamique sur un seul graphe. L'idée principale est d'associer à chaque arête un domaine temporel d'existence. En représentant un graphe où tous les nœuds partageant un lien à quelque moment que ce soit sont reliés par une arête, on peut étiqueter les arêtes avec leur domaine temporel d'existence. On peut toujours convertir un graphe évolutif de manière à obtenir un TVG sur un graphe, comme le montre par exemple la figure 4.5.

Représenter le TVG sur un seul graphe peut faciliter la reconnaissance des chemins temporels, puisqu'il suffit d'essayer de suivre un chemin du graphe en choisissant à chaque arête une date de son étiquette postérieure à celle de l'arête précédente. Lorsque les changements temporels sont nombreux (par exemple sur une longue durée d'observation), cette représentation peut également être plus facile à lire qu'une longue suite de graphes. Cependant, si le réseau est dense, dans chaque topologie, le degré des nœuds est plutôt élevé et on aboutit rapidement à un graphe complet qui peut être difficile à lire correctement en raison d'étiquettes très fournies.

Un modèle équivalent aux TVG est proposé par [**?**] est nommé *stream graph*, proposant notamment une prise en compte spécifique de la disparition ou l'apparition d'un nœud dans le réseau. Ce travail vise notamment à étendre les propriétés de la théorie des graphes (statiques) à des graphes dynamiques.

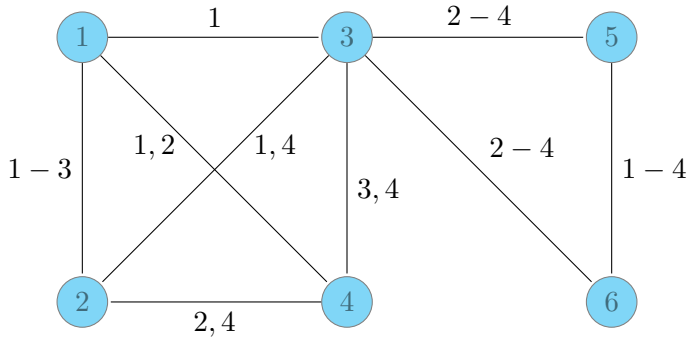
4.3.1 Classification de topologie dynamique

Un TVG (tout comme un stream graph) doit permettre l'étude de propriétés topologiques au cours du temps malgré les changements topologiques et la classification des topologies dynamiques selon leurs propriétés. Grâce au concept de chemin temporel, il est possible d'étendre les propriétés de la théorie des graphes concernant les chemins.



$t = 1$ $t = 2$ $t = 3$ $t = 4$

Observation de la dynamique du réseau



Time Varying Graph (TVG))

FIGURE 4.5 – Exemple de topologie dynamique représentée par une séquence de graphes (Observation de la dynamique du réseau) puis à l'aide d'un TVG représenté sur un graphe unique. Toutes les arêtes apparaissant dans l'un des graphes de l'observation apparaissent dans le TVG avec une étiquette indiquant leurs dates de présence.

Un chemin temporel est dit direct lorsque chaque arête qui le compose est utilisable immédiatement après la précédente. Une fonction (nommée ζ) attribue une latence à chaque couple arête-date indiquant la durée d'un transfert sur l'arête e à la date t. Dans le cas trivial où $\zeta(e, t) = 0$ (quelles que soient l'arête e et la date t), l'existence à une date t' un chemin temporel direct signifie que toutes les arêtes du chemin sont présentes à la date t'. Dans un réseau dynamique, l'existence d'un chemin temporel direct permet le transfert d'une communication d'un bout à l'autre du chemin, même si les nœuds du réseau n'ont pas connaissance de la topologie s'ils retransmettent tous la communication reçue sans attendre. Lorsqu'un chemin temporel n'est pas direct, il est dit indirect. Dans ce cas, une certaine connaissance de la topologie s'avère nécessaire pour faire transiter des données sur le chemin temporel.

La « vision » d'un nœud u sur un autre v (*temporal view*) à la date t détermine la date de départ la plus récente d'un chemin temporel au départ de v arrivant à u au plus tard à la date t. La vision d'un nœud sur un autre définit donc l'état le plus récent de l'autre nœud dont il peut avoir connaissance.

Puisque les chemins temporels sont à la fois caractérisés par leur distance topologique (le nombre d'arêtes qui les constituent) mais aussi par leur durée (distance temporelle), un TVG dispose à la fois d'un diamètre topologique (en nombre d'arêtes) et temporel (en durée). La connexité d'un TVG est héritée des définitions ayant cours pour

les graphes orientés, même si les liens du TVG sont non orientés, parce que les chemins temporels le sont. On peut ainsi déterminer les composantes fortement et faiblement connexes d'un TVG comme le montre [**?**].

À l'aide de ces propriétés, il devient possible de classifier les TVG, ce qui pourrait permettre la caractérisation de réseaux dynamiques réels comme proposé dans [**?**]. La recherche dans le TVG de chemins temporels satisfaisant certaines propriétés permet de définir des propriétés du TVG qui sont ponctuelles, c'est-à-dire qu'un TVG entre dans la classe même si cette propriété est uniquement satisfaite à un instant particulier de l'observation. Un TVG dans lequel un nœud est le départ de chemins temporels menant à chacun des autres nœuds fait partie de la classe des TVG à source de donnée (*data source*), tandis que s'il est l'arrivée de chemins temporels au départ de chacun des autres nœuds, il s'agit d'un puits de données (*data sink*). Un TVG est temporellement connexe si chaque nœud est une source de donnés tandis qu'il assure la connexité avec retour si tout couple de nœuds fait partie d'un circuit temporel.

Lorsqu'on s'intéresse à un TVG infini (ou, par approximation, très long), il peut être intéressant de rechercher des propriétés récurrentes. La connexité récurrente est assurée si après chaque instant, il existe un chemin temporel entre chaque couple de nœud. La récurrence des arêtes consiste, pour chaque arête du TVG, à ne jamais disparaître définitivement.

Assurer certaines propriétés à tout instant du TVG permet d'étudier certaines exécutions sans considérations pour les dates des configurations atteignables. Ainsi, une période bornée de récurrence des arêtes permet d'assurer qu'un nœud pourra communiquer avec son voisin en une durée maximale connue. Elle permet également d'assurer que le diamètre temporel soit également borné. Une propriété intéressante est alors la connexité topologique permanente, soit le fait que la topologie instantanée à chaque date soit connexe. Sa variante au cours du temps est l'existence à chaque instant et pour chaque couple de nœuds du réseau d'un chemin temporel au départ de l'un et à l'arrivée de l'autre. Cette propriété est également nommée routabilité à terme car l'existence à chaque instant de tels chemins temporels assurerait à un nœud disposant d'un message à destination d'un autre de pouvoir, sous réserve de disposer de la topologie à venir du réseau, router ce message jusqu'à sa destination immédiatement.

Le modèle des TVG a été utilisé notamment utilisé dans l'étude des réseaux dynamiques dont font partie les réseaux véhiculaires. En effet, dans [**?**], les TVG sont utilisés notamment pour caractériser un algorithme mesurant la latence dans un réseau dynamique tolérant aux délais (*DTN*). Le routage dans de tels réseaux est étudié via ce modèle dans [**?**], par exemple. Les réseaux tolérants aux délais peuvent parfois désigner des réseaux non informatiques, comme des réseaux sociaux. Les TVG sont également utilisés dans ce contexte comme par exemple dans [**?**, **?**, **?**, **?**]. L'étude du routage en réseau dynamique fait également usage de ce modèle, notamment dans [**?**], dans un réseau dynamique qui ne constitue pas forcément un DTN.

4.4 Modèle des p-graphes dynamiques

En surcouche des graphes dynamiques, on peut simplifier la gestion du temps en prenant en compte des paramètres techniques qui permettent une certaine discrétisation sans trop de pertes d'informations. On les construit grâce à une fonction nommée δ qui encapsule les paramètres réseau, et un graphe dynamique représentant l'observation de

la dynamique du réseau. Le modèle des p-graphes dynamiques proposé par [**?**] cherche à la fois une meilleure prise en compte des paramètres techniques sous-jacent aux communications et une prise en compte de la dynamique à différentes échelles de temps.

4.4.1 Motivation

L'étude d'un réseau dynamique basé sur un protocole de communication sans fil nécessite de prendre en compte à la fois les changements topologiques et les paramètres du protocole de communication.

En effet, l'existence de deux nœuds reliés par un chemin temporel ne garantit pas toujours qu'une communication utile puisse être réalisée depuis le nœud de départ vers le nœud d'arrivée. Il est par exemple possible, si l'une des arêtes du chemin temporel est trop courte pour transmettre un paquet (PDU du protocole) entier, que ce chemin ne soit pas exploitable. Les paramètres techniques (débit, délais d'attente, etc.) des communications doivent donc constamment être pris en compte lorsqu'on analyse un réseau dynamique sans fil comme un réseau véhiculaire.

La prise en compte de ces deux réalités implique, pour chaque analyse de chemin temporel, une étude fastidieuse des dates et durées de chaque arête qui le constitue à l'aune des paramètres technologiques. C'est ainsi que [**?**] propose le modèle des p-graphes dynamiques, qui rend compte de manière synthétique des capacités d'échanges réelles au sein d'un réseau de topologie dynamique.

4.4.2 Construction

Le modèle des p-graphes dynamiques est basé sur une observation. Dans cette observation, chaque lien a une durée, qui affecte la quantité de données transmissibles. Dans les réseaux sans fil, et en particulier en contexte véhiculaire, les systèmes de communication utilisent généralement un format de communication prédéterminé incluant une taille de messages normée ([**?**, **?**, **?**] pour des messages véhiculaires).

Le transfert d'un message a donc, dans ces circonstances, une durée qui peut être prédite, en utilisant les caractéristiques techniques du système de communication (bande passante, délais protocolaires). À toute observation, il est alors possible d'associer une fonction de durée de transfert que les auteurs nomment $\delta : \mathbb{N}^* \to \mathbb{R}^+$. Cette fonction affecte à chaque entier $p \geq 1$ une durée suffisante pour transférer p messages successifs. Cette fonction est considérée identique quel que soit le nœud émetteur, et lorsque les messages sont de taille variable, il est possible de la déterminer en prenant la taille maximale de message.

À partir de l'observation et de la fonction de durée de transfert utilisée, il est possible, pour une valeur donnée de p, de calculer à tout moment le p-graphe correspondant à l'observation selon la méthode décrite par [**?**]. Pour cela, il suffit, à chaque instant t en partant du début de l'observation, de sélectionner les liens existant depuis la date t et jusqu'à la date $t + \delta(p)$. Ils forment alors le p-graphe qui est ajouté à la fin du p-graphe dynamique $\mathcal{G}^p$ s'il n'est pas identique au précédent. Le p-graphe dynamique $\mathcal{G}^p$ comprend tous les p-graphes représentant l'observation depuis son démarrage mais aucun couple de p-graphes consécutifs de $\mathcal{G}^p$ n'est composé de deux p-graphes identiques. L'ordre des p-graphes dans $\mathcal{G}^p$ respecte la chronologie des changements topologiques. Ce p-graphe dynamique représente alors tous les changements topologiques à prendre en compte lorsqu'on étudie l'émission de p messages consécutifs.

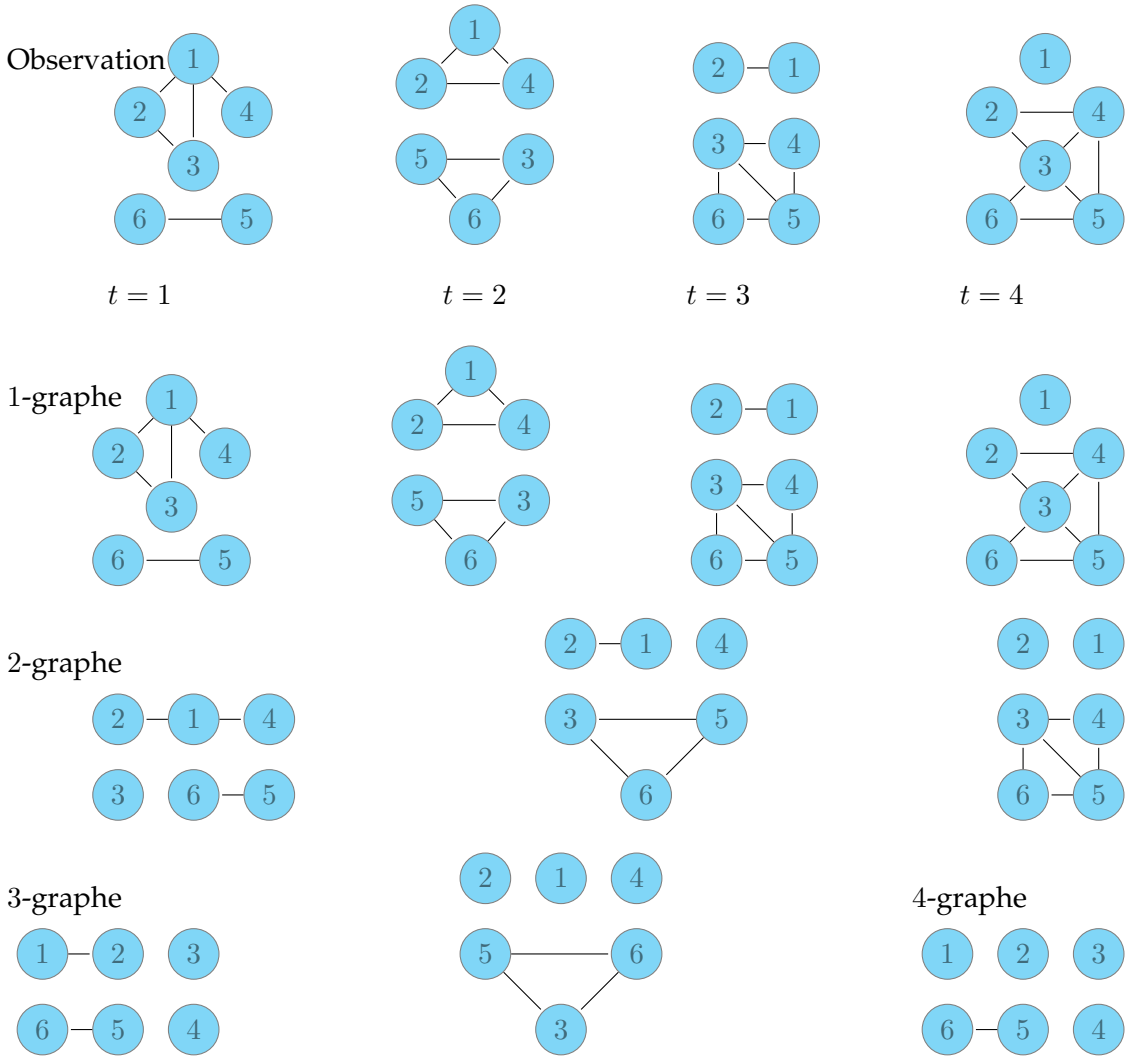


FIGURE 4.6 – Exemple de construction d'une famille de p-graphes $\mathcal{F}$ à partir d'un TVG, en utilisant la fonction de durée de transfert simple $\delta(p) = p$. Les dates disparaissent lors de la construction d'un p-graphe dynamique mais l'ordre des p-graphes qui le composent reste chronologique.

On peut répéter la construction d'un p-graphe dynamique pour toute valeur entière de p (supérieure à 1). Ce processus est appliqué à la figure 4.6 sur la même observation que celle étudiée à la figure 4.5. On obtient alors une famille de p-graphes dynamiques $\mathcal{F} = (\mathcal{G}^p)_{p\in\mathbb{N}^*}$, représentant l'évolution topologique à prendre en compte quel que soit le nombre de message consécutifs p auquel on s'intéresse (il existe toujours dans la famille, un p-graphe dynamique pour cela). Pour chaque observation, il existe une valeur de p à partir de laquelle [**?**] a montré que tous les p-graphes dynamiques sont identiques. On peut donc résumer la famille en une collection d'un nombre fini de p-graphes dynamiques.

Enfin, un p-graphe dynamique s'abstrait des connexions courtes (dont la durée est inférieure à $\delta(p)$), ce qui peut permettre de résumer les changements topologiques d'un réseau. Cette application se sert du modèle pour atténuer le « bruit » dans la topologie

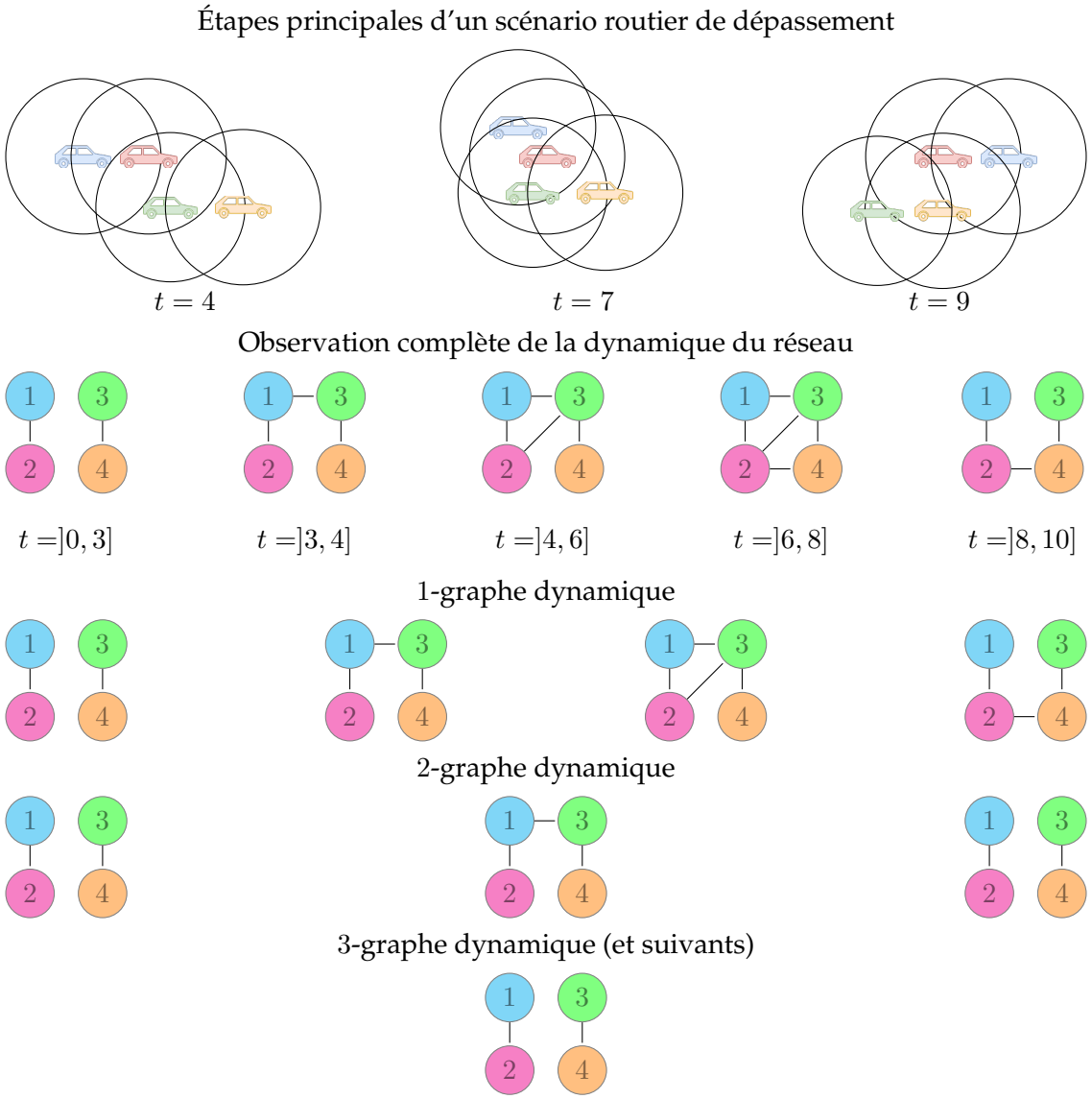


FIGURE 4.7 – Exemple d'utilisation d'une famille de p-graphes pour résumer les changements topologiques d'un réseau dynamique en supprimant le bruit lié aux interactions brèves. La durée de transmission de p messages est $\delta(p) = 2 \times p$.

dynamique causé par des liens courts entre des nœuds qui s'avéreraient difficile à utiliser pour une coopération efficace. Un exemple de topologie dynamique résumée dans le cas d'un dépassement (sur autoroute, par exemple) est proposé à la figure 4.7.

4.4.3 Limites de la modélisation

La modélisation d'une observation sous forme de p-graphes n'est cependant pas parfaite. En effet, elle rend par exemple plus difficile l'analyse de la durée exacte des liens (la discrétisation peut faire perdre de l'information sur l'observation si les dates sont effacées). Il s'agit également d'un modèle dont la pertinence peut s'avérer réduite lorsqu'on s'intéresse aux interactions sur de longues distances topologiques (routage global sur un

grand réseau dynamique) ou de longues durées (réseaux tolérant les délais ou *DTN*). En effet, étudier des échanges séparés par une longue durée fera intervenir des arêtes apparaissant dans des p-graphes différents, séparés par une séquence (éventuellement très longue) de p-graphes inexploités. Par ailleurs, si les liens de communications sont supposées longs par rapport aux transferts à effectuer, le nombre précis de messages échangeables est une donnée qui perd de l'importance.

Lorsque les paramètres technologiques ou algorithmiques (nécessaires à établir la fonction de durée de transfert) sont difficiles à déterminer (par exemple lorsque la taille des messages est très variable), le modèle perd également de l'information par rapport à un modèle direct de l'observation. Malgré cela, les réseaux véhiculaires concernent généralement des interactions temporellement rapprochées entre nœuds spatialement proches. Si les messages des protocoles de coopération véhiculaires sont de taille variable, il est généralement aisé de déterminer un maximum proche des usages réels et la caractérisation de la distance topologique est de première importance (échanges à n sauts, notamment). C'est pourquoi le modèle des p-graphes dynamiques semble adapté à l'étude des réseaux véhiculaires.

4.4.4 Choix et application du modèle

Dans un réseau véhiculaire, les informations peuvent rapidement devenir obsolètes. C'est par exemple le cas de la position des véhicules, des obstacles qu'ils détectent ou des décisions qu'ils prennent. C'est pourquoi l'existence d'un chemin temporel entre deux véhicules n'a généralement d'intérêt que lorsque sa durée est très faible, ce qui n'est pas toujours facile à repérer dans un graphe dynamique (comme un TVG, par exemple).

Les véhicules intéressés par une donnée sont généralement géographiquement très proches de l'emplacement auquel cette donnée a été générée. En effet, il s'agit généralement de données qui peuvent être comparés avec des perceptions du véhicule (donc limitées par la portée des capteurs), ou qui servent à déterminer les risques de collision imminente (insignifiants à trop grande distance) ou encore à des prises de décisions collaboratives concernant des ressources à partager (portions de routes, d'intersections), qui ne concernent que les véhicules qui en sont proches. Les données concernant des lieux géographiquement éloignés du véhicule (et qu'il ne visitera peut-être même pas) n'ont donc habituellement pas d'intérêt pour ce dernier.

Les nœuds du réseau véhiculaire cherchent ainsi à disposer d'informations récentes concernant leur environnement géographique proche. Dans ces circonstances, un chemin temporel se doit à la fois d'être court et rapide (*short* et *fast*) pour présenter un intérêt dans la communication. Le modèle, des p-graphes dynamiques, qui rend compte de ces deux caractéristiques en même temps pourrait donc s'avérer utile pour faciliter l'étude des réseaux dynamiques de véhicules.

Lorsque la fonction de durée de transfert utilisée pour construire une famille de p-graphes est proportionnelle ($\delta(p) = p \times \delta(1) \forall p \in \mathbb{N}^*$), la famille donne à la fois des indications sur le nombre de messages qui peuvent être transmis (une p-arête permet d'échanger p messages consécutifs) et sur la distance à laquelle un message peut être transmis. En effet, tous les liens d'un p-graphe durent suffisamment longtemps pour que p nœuds consécutifs émettent un message l'un après l'autre, par exemple en faisant suivre le message précédent ou en y ajoutant des données (*piggy-backing*). Dans la suite de ce travail, seules des fonctions de durée de transfert proportionnelles seront étudiées.

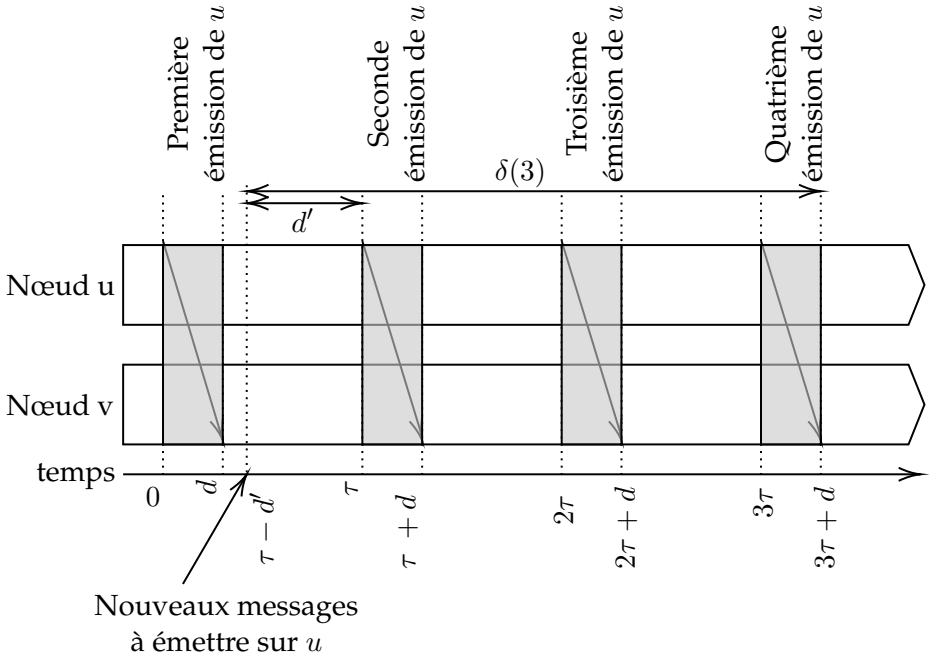


FIGURE 4.8 – Schéma illustrant l'expression de la fonction δ. Ici, la transmission du paquet de 3 messages dure $\delta(3) = 3\tau + d - (\tau - d') = 2\tau + d + d'$

Si la fonction de durée de transfert δ dépend des paramètres du protocole de communication, elle est également considérée dépendante de l'algorithme réparti étudié dans ce travail. En effet, c'est l'algorithme qui détermine la taille du message (au travers de son contenu et son formattage), et les éventuels durées d'attentes incompressibles (au travers de ses spécifications, comme on peut le trouver dans les normes sur les messages CAM ou CPM par exemple). Ainsi, le choix d'une fonction de durée de transfert non proportionnelle rendrait la prise en compte de ces paramètres complexes.

Lorsque l'algorithme de coopération est basé sur des émissions périodiques, comme c'est souvent le cas des algorithmes envisagés dans un réseau véhiculaires (c'est notamment le cas des normes [**?**, **?**, **?**]), cette période τ est impliquée dans la définition de la fonction δ. Le schéma de la figure 4.8 représente cette relation. Au sein d'un algorithme à émissions périodiques, l'émission de 3 messages est déclenchée sur le nœud u à une date quelconque entre deux émissions. Le premier message sera transmis avec l'émission suivante, et ainsi de suite jusqu'à ce qu'il n'y en ait plus. Chaque période d'émission est dénotée en gris et par une flèche oblique reliant u à v. Les dates des différents événements sont indiqués en bas du schéma. Il s'agit des dates perçues par un observateur omniscient extérieur au système.

En utilisant le schéma de la figure 4.8, on peut déduire que la fonction de durée de transfert peut, lorsque les transmissions sont basées sur des émissions périodique, s'exprimer sous la forme $\delta : p \to \tau \times (p-1) + d' + d$. La valeur de d n'est dépendante que du protocole de communication, on peut caractériser la valeur de d' de la façon suivante $d' \in [0, t - \delta(1)]$. Par approximation, et puisque cela représente un pire cas, nous considérerons dans ces circonstances que la fonction δ s'exprime de la façon suivante : $\delta(p) = p \times \tau$ $(\forall p \in \mathbb{N}^*)$.

4.5 Comparaison des modèles

Les modèles de topologie développés dans ce chapitre peuvent tous permettre de caractériser la dynamique supportée par un algorithme réparti comme l'algorithme RDF (algorithme 3 détaillé à la section 3.4). Nous proposons dans cette section d'esquisser les étapes nécessaires pour démontrer cet algorithme dans un environnement dynamique modélisé avec deux des modèles de topologie réseau dynamique identifiés dans ce chapitre.

Dans les deux cas, on supposera que les nœuds détectant qu'un message doit être émis pour combler le retard d'un de leurs voisins l'émettent aussitôt.

Si la topologie réseau dynamique est modélisée à l'aide d'un TVG, on suppose l'existence d'un circuit temporel direct composé de 2 arêtes qui part d'un nœud retardataire et y revient via un nœud à jour. Dans ces conditions, le circuit temporel direct permet, si le nœud retardataire émet à la date à laquelle il existe, l'échange de messages nécessaire au rattrapage du retard.

En utilisant le modèle des p-graphes dynamiques, il suffit de considérer le 2-graphe représentant l'observation dans une configuration du système où le nœud retardataire émet un message quelconque. S'il est voisin d'un nœud à jour dans le 2-graphe, l'échange de messages nécessaire au rattrapage est possible.

Les deux modèles permettent de poser des conditions sur la topologie dynamique du réseau, assurant le fonctionnement de l'algorithme RDF. L'utilisation du modèle des p-graphes dynamiques permet de faire abstraction du temps à l'intérieur de la démonstration, en n'utilisant que la suite logique des configurations d'une exécution, tandis que l'utilisation du modèle des TVG s'est basée sur la comparaison de dates.

Enfin, l'expression des propriétés topologiques semble simplifiée par l'utilisation du modèle des p-graphes dynamiques. En effet, l'existence d'une arête dans le 2-graphe permet à un nœud de rattraper son retard, tandis que dans le TVG, il s'agit de l'existence d'un circuit temporel direct de 2 arêtes.

Ces éléments permettront de rendre la démonstration utilisant les p-graphes dynamiques plus lisible, et plus proche des démonstrations habituelles en algorithmique répartie.

4.6 Conclusion

La topologie d'un réseau dynamique change constamment ce qui la rend plus difficile à modéliser que celle d'un réseau statique. La communauté scientifique a établi des formalismes basés sur des graphes, permettant de modéliser des topologies dynamiques.

De telles modélisations permettent notamment d'exprimer des propriétés d'une topologie dynamique dans laquelle le fonctionnement d'un algorithme est démontré. Parmi ces modélisations, les TVG sont des graphes constitués d'arêtes datées. La plupart des propriétés des graphes peuvent être adaptées à des TVG, mais ne peuvent pas directement être considérées. Une autre modélisation, nommée famille de p-graphes dynamiques, se base sur des p-graphes, dont les arêtes caractérisent les liens qui durent suffisamment longtemps pour transmettre p messages.

L'utilisation de ces deux modèles de topologie dynamique pour démontrer l'algorithme RDF vu au chapitre précédent a permis d'illustrer leur potentiel. Le modèle des p-graphes dynamiques semble particulièrement adapté pour exprimer des propriétés

plus générales d'une topologie réseau de manière simple et facile à vérifier. C'est le modèle qui sera utilisé dans le reste des travaux de thèse pour modéliser une topologie réseau dynamique. Toute utilisation de ce modèle pourrait être remplacée par l'utilisation d'un autre modèle de topologie dynamique tel que les TVG, mais cela peut compliquer l'expression des propriétés topologiques et les rendre plus difficiles à comprendre et à vérifier.

Le modèle de topologie dynamique choisi est alors utilisé afin d'élaborer une méthode de prédiction de performances d'une application de coopération en réseau dynamique de véhicules.

Chapitre 5

Méthode de prédiction de comportement algorithmique

Sommaire

5.1 Introduction

Après avoir abordé les modélisations utiles pour l'étude formelle d'algorithmes de coopération véhiculaire, il devient désormais possible d'élaborer une méthode de prédiction de performances adaptée aux contraintes des réseaux véhiculaires.

L'environnement routier est sujet à de nombreuses règles, comme nous l'avons vu au chapitre 1, à la fois pour des raisons de sécurité routière mais aussi pour assurer la responsabilité civile de la réparation des dommages subis sur route. Dans ce contexte, si la coopération entre véhicules a de grandes applications pour améliorer la sécurité routière et la fluidité du trafic, cela n'est pas sans poser de problèmes tant au niveau technique que concernant les responsabilités légales. Comme nous l'avons vu au chapitre 2, le bon fonctionnement de telles applications est incertain ce qui pourrait être à l'origine de décisions causant des dommages. Dans ces conditions, le développeur de même que

l'exploitant (par exemple le gestionnaire d'un réseau routier) prennent le risque de voir leur responsabilité engagée.

Une approche de validation hybride est proposée dans ce chapitre, alliant à la fois une analyse formelle de l'algorithme pour plus de garanties, mais aussi une étude empirique d'observations pour plus de réalisme. Elle a vocation à être suffisamment simple à mener pour pouvoir être utilisée en amont des déploiements réels, et que son utilisation à chacun d'entre eux soit automatisable. Son réalisme doit également être suffisant pour que les garanties fournies soient utilisables dans un contexte véhiculaire réel. Cela signifie à la fois que les hypothèses sur la dynamique doivent pouvoir être réunies dans des réseaux véhiculaires réels, mais également qu'il est possible de vérifier simplement qu'elles sont réunies.

Dans ce chapitre, une étude des méthodes de validations employées pour les applications de coopération véhiculaire est tout d'abord menée, afin de positionner la méthode proposée. Chaque étape de cette méthode est ensuite détaillée, en commençant par l'étude algorithmique, suivie par l'étude d'observations. Des conclusions et remarques clôturent enfin le chapitre.

5.2 Validation d'applications véhiculaires coopératives

Il existe plusieurs méthodes permettant la validation d'une application répartie destinée à un contexte véhiculaire. Certaines sont empiriques, basées sur des essais expérimentaux qui peuvent être partiellement ou totalement simulés, et d'autres plus théoriques, basées sur des modèles et leurs propriétés. En étudiant, pour chacune, les limites et intérêts spécifiques, il devient possible de proposer une méthode de validation adaptée au contexte véhiculaire et à l'exigence de garanties par les différents acteurs en situation réelle.

5.2.1 Validation par des tests routiers

Pour assurer le bon fonctionnement d'une application de coopération véhiculaire en situation réelle, la méthode la plus intuitive est de tester l'application dans un déploiement réel. Pour cela, l'application est implémentée et exécutée dans l'infrastructure routière et différents véhicules effectuent sur route des trajets réalistes représentatifs des différentes situations réellement rencontrées dans le contexte visé.

Si les tests sont trop peu nombreux ou trop peu représentatifs du trafic réel, il peut être difficile d'inférer des garanties de bon fonctionnement en situation réelle. En effet, il reste possible que les tests se soient concentrés sur des situations particulières parmi les plus favorables au fonctionnement de l'application de coopération. Une fois déployée, l'application pourrait fonctionner différemment et provoquer des accidents ou ne plus atteindre des performances satisfaisantes.

Dans les faits, les tests routiers consomment énormément de ressources. Il s'agit notamment de ressources humaines, par exemple pour conduire les véhicules, installer le matériel et les logiciels utilisés, coordonner les conducteurs et récupérer les données. Les ressources matérielles ne sont pas non plus à négliger, notamment les véhicules, l'espace routier et les équipements embarqués pour la communication et l'exécution de l'application.

En raison de ces coûts et des défis logistiques conséquents posés par les expérimentations réelles, il n'est pas rare de voir les expérimentateurs réduire le spectre des expériences, par exemple en limitant le nombre et la représentativité des tests ou bien en limitant le nombre de véhicules impliqués.

Par exemple, les auteurs de [**?**] proposent une application de coopération entre piétons et véhicules qu'ils valident avec des expériences réelles. Leur application est conçue pour prévenir les collisions entre véhicules et piétons par des communications entre les véhicules et le smartphone des piétons. Les expériences de validation se déroulent dans un environnement routier (parking) mais n'impliquent jamais plus d'un véhicule et un piéton. Même si les expériences sont répétées et les paramètres du scénario sont variés (vitesse des véhicules, notamment), le nombre de véhicules et piétons impliqués et le cadre routier ne sont pas forcément représentatifs d'un éventuel déploiement réel. Le nombre de piétons et véhicules impacte directement la quantité d'émissions radio et donc les performances réseau du système proposé dans son ensemble.

Un autre exemple est proposé dans [**?**] où les auteurs décrivent une architecture de perception collaborative utilisant un protocole de communication de leur conception. Cette architecture est testée sur route dans des scénarios impliquant 2 véhicules collaborant pour en détecter un troisième. Les expériences réalisées n'étant pas représentatives d'une situation réelle, ils ne les utilisent qu'en tant que Preuve de Concept (PoC), qui serait clairement insuffisante pour un déploiement réel.

En réalité, la plupart des validations expérimentales ont généralement une valeur assumée de démonstration conceptuelle, comme par exemple dans [**?**, **?**, **?**], où le nombre d'expériences et le contexte des tests (circuit fermé, nombre de piétons et autres véhicules réduits) empêche les résultats d'être représentatifs d'un déploiement réel.

Même ces validations conceptuelles sont coûteuses à la fois d'un point de vue financier et logistique, raison pour laquelle certaines preuves de concept ne font pas appel à des véhicules mais des miniatures robotisées [**?**, **?**]. Ces petits robots, peu coûteux et faciles à déployer sont normalement mobiles et capables de communiquer. Certaines preuves de concept font même l'impasse sur la mobilité, de toutes façons peu représentatives des déplacements réels des véhicules comme [**?**] où l'expérience est réalisée sur des Raspberry Pi fixes.

Si les expériences réelles sont importantes pour valider le concept (assurer une implémentation complète, l'interfaçage avec le protocole de communication, démontrer la faisabilité), il est généralement trop complexe de mettre en place une expérience réellement représentative des conditions réelles dont les résultats pourraient être (même partiellement) extrapolés à un futur déploiement. La valeur prédictive des expériences en conditions réelles est généralement trop faible pour assurer les garanties de bon fonctionnement préalable à tout déploiement réel.

5.2.2 Validation par des preuves algorithmiques

Il est possible de s'épargner les difficultés des expériences réelles en adoptant une approche plus théorique de la validation. En effet, une preuve théorique du bon fonctionnement d'un algorithme réparti de coopération pour des réseaux dynamiques assurerait son bon fonctionnement réel.

La démonstration d'un algorithme réparti repose toujours sur un modèle du système réparti dans lequel il s'exécute. Ce modèle définit notamment comment sont effectués les calculs locaux et comment sont réalisées les communications (modèles synchrones, asyn-

chrones). Il définit également précisément quelles pannes peuvent survenir (pannes byzantines, arrêt de processus, pertes de messages). Les hypothèses définissant le modèle peuvent être irréalistes (actions atomiques, arrêt de processus sans redémarrage possible) ou difficiles à vérifier dans un réseau véhiculaire (synchronisation parfaite, équité des pertes de messages). Ainsi, la plupart des preuves algorithmiques n'offrent que peu de garanties sur le bon fonctionnement de l'algorithme en conditions réelles (les hypothèses sont souvent inapplicables ou leur applicabilité est difficile à vérifier).

Même avec des hypothèses très restrictives sur le système réparti, il est impossible de démontrer le bon fonctionnement d'un algorithme réparti pour tous les types de réseaux dynamiques. En effet, un réseau suffisamment mauvais pour empêcher tous les messages d'être transmis (mobilité trop grande, canal de communication trop lent ou avec trop d'interférences) assure, quoi qu'il arrive localement, que l'algorithme réparti échouera. Les démonstrations d'algorithmes répartis en réseaux dynamiques doivent donc faire des hypothèses sur la dynamique du réseau et la modéliser.

La modélisation d'un réseau statique se fait habituellement sous forme de graphes, dont les nœuds représentent les machines du réseau tandis que les arêtes représentent les liens de communications disponibles. Dans le cadre d'un réseau dynamique, il est classique d'utiliser un formalisme de graphes dynamiques capable de montrer les changements topologiques. Comme le temps est une donnée cruciale pour la dynamique du réseau, il est possible de dater les changements topologiques mais aussi de prendre en compte la latence de transmission et les paramètres du canal de communication (protocole de communication et canal physique de transmission).

On attend généralement d'un algorithme réparti destiné à un réseau dynamique d'être capable de fonctionner à la fois malgré des pertes de messages à la fois causées par les changements topologiques, mais aussi par l'environnement électromagnétique (collisions et interférences). En effet, d'un point de vue algorithmique, une instance locale de l'algorithme est incapable de différencier ces deux situations. Ainsi, les démonstrations d'algorithmes répartis en réseaux dynamiques considèrent généralement toutes les pertes de messages de la même manière. C'est notamment l'approche utilisée dans [**?**], où les auteurs tentent de démontrer divers algorithmes capables de créer et maintenir des arbres couvrants au sein d'un réseau à topologie dynamique. Avec cette approche, on ne peut fournir de garanties de bon fonctionnement en situation réelle que dans la mesure où l'on peut prédire les pertes de messages liées à l'environnement électromagnétique, ce qui semble particulièrement difficile à réaliser.

Même en réseau statique, la démonstration d'un algorithme réparti peut être difficile à obtenir, et, dans certains cas, impossible comme démontré dans [**?**]. Dans un réseau dynamique, la démonstration est d'autant plus complexe ce qui limite son champ d'application. Lorsqu'une preuve semble difficile à obtenir, il est généralement nécessaire de faire des hypothèses simplificatrices sur la dynamique du réseau. Dans [**?**], les auteurs étudient des algorithmes auto-stables dans des réseaux dynamiques. Pour démontrer les algorithmes, ils formulent ces hypothèses sous la forme de classes de graphes dynamiques.

Lorsque les hypothèses simplificatrices sur la dynamiques sont choisies pour simplifier la démonstration, elles peuvent manquer de réalisme compte tenu du contexte d'usage. Dans ces situations, la démonstration n'apporte plus aucune garantie pour une situation réelle. En effet, si la situation réelle ne garantit pas l'une des hypothèses de la preuve, sa conclusion n'y est pas applicable. C'est ainsi que la plupart des preuves al-

gorithmiques en réseau dynamique ne permettent pas d'offrir de garanties en situations réelles, leurs hypothèses n'étant pas suffisamment réalistes ou trop difficiles à vérifier en réalité dans un système aussi complexe.

5.2.3 Validation par simulation ou émulation

Lorsque les scientifiques étudient des systèmes complexes sur lesquels ils ne peuvent pas expérimenter de manière réaliste (le climat, les réacteurs nucléaires, les systèmes biologiques, les accidents routiers) ils utilisent souvent des simulations. Cette approche permet à la fois d'éviter les difficultés éthiques, logistiques et financières des expérimentations réelles, mais aussi d'étudier des systèmes qui n'existent plus (archéologie, climatologie), ou pas encore (recherche et développement). Dans le contexte des réseaux véhiculaires, cela permet d'étudier la faisabilité et l'intérêt d'un déploiement sur une route existante, mais aussi sur un projet à venir (autoroute en construction, modification de carrefour) afin d'intégrer dès la conception du projet les fonctionnalités liées à la coopération véhiculaire.

La simulation de réseaux de communications est facilitée par l'existence d'outils très connus (simulateurs de réseau NS [**?**], OMNET++ [**?**], the ONE [**?**], simulateur de trafic routier SUMO [**?**]) pour lesquels la communauté a également développé des extensions variées pour différents cas d'usage. Grâce à eux, on peut effectuer de multiples tests dans des configurations voisines ou très différentes, de manière à pouvoir plus facilement extrapoler les résultats à une situation réelle. Ils permettent également l'optimisation des paramètres expérimentaux, ce qui peut aider à améliorer les performances d'un déploiement.

Les simulations de réseaux véhiculaires reposent sur de nombreuses couches simplificatrices de modélisation, dont l'impact sur le réalisme du résultat n'est pas forcément neutre. Une expérience de simulation d'expérience de coopération véhiculaire nécessite généralement :

— la simulation du trafic routier ;

— la simulation de l'implémentation du protocole de communication (généralement fortement simplifiée par rapport à une implémentation réelle) ;

— la simulation de l'application véhiculaire (l'implémentation pour le simulateur diffère généralement sensiblement d'une implémentation réelle) ;

— la simulation du trafic de données sur le réseau.

Ces simplifications et leurs interactions peuvent avoir un impact significatif sur le résultat, comme montré dans [**?**] pour des réseaux non véhiculaires. Du fait d'une mobilité plus importante dans le cadre routier, et qui interagit donc plus avec les autres composantes simulées, ces difficultés ne peuvent être qu'amplifiées [**?**] dans le cas de réseaux véhiculaires. Certains travaux comme [**?**] montrent d'ailleurs des divergences significatives sur la portée effective entre les résultats obtenus à travers deux simulateurs différents (NS-3 et omnet++). Cette divergence est observée dans des scénarios très simples (impliquant 2 véhicules s'échangeant uniquement des messages périodiques BSM), elle peut mener à de grandes différences de comportement lors de l'utilisation de scénarios et d'algorithmes de coopération plus complexes. Une comparaison avec des tests routiers réels montre que la différence est, pour certaines valeurs de modulation, encore plus grande avec le test réel qu'avec les simulations.

Pour améliorer le réalisme des simulations, il est possible de limiter le nombre d'éléments simulés. Dans une émulation, les communications sont virtuelles tandis que les autres composants sont réels. Pour un réseau véhiculaire, cela signifie que le comportement de chaque véhicule est calculé séparément, et que l'application testée est implémentée à l'identique d'un déploiement réel. Il est ainsi possible de diminuer les possibles écarts avec la réalité. De plus, disposer d'une implémentation fonctionnelle de l'application facilite la mise en place de preuves de concept réelles et fonctionnelles. On peut par exemple utiliser des données d'un test routier réalisé en amont, notamment une capture du bus CAN du véhicule, de sa position GPS et des interactions avec l'utilisateur. Il est également possible d'ajouter des interactions avec un humain au cours de l'émulation puisque l'application s'exécute comme en conditions réelles.

Calculer séparément le comportement de chaque application dans chaque véhicule implique une consommation de ressources (capacités de calcul, mémoire, temps, etc.) d'autant plus forte que le réseau émulé comporte de nombreux composants. La consommation de ressources peut donc empêcher d'expérimenter lorsque le réseau à émuler est trop grand ou complexe. Réduire l'échelle de l'expérience peut alors être nécessaire mais avoir un fort impact sur le réalisme du test.

L'inclusion dans les expériences d'éléments réels rend plus difficile la création d'outils standards et l'on assiste souvent à l'utilisation d'un outil différent chez chaque équipe de recherche. Par exemple, des outils d'émulation de réseaux véhiculaires différents sont proposés dans [**?**] et [**?**], cherchant à chaque fois un meilleur degré de réalisme. En effet, comme le montre [**?**], l'émulateur peut être paramétré avec des valeurs mesurées dans des tests routiers réels comme le taux de pertes, ou la latence, tandis que [**?**] propose de prendre en compte des éléments physiques comme la propagation du signal à travers l'effet Doppler.

Les difficultés à émuler un réseau dynamique de grande taille ont poussé les auteurs de [**?**] à proposer une approche intermédiaire où seuls certains nœuds sont complètement émulés tandis que d'autres sont simulés. Le choix des nœuds à émuler est alors de première importance puisqu'il définit quels nœuds ont le comportement le plus réaliste.

L'émulation est l'une des méthodes de validation les plus utilisées, par exemple dans [**?**, **?**, **?**, **?**], malgré le manque d'outils standards. Elle est appréciée pour son réalisme généralement accru par rapport à une simulation complète. La consommation parfois importante de ressources empêche cependant d'obtenir des résultats aussi réalistes par cette approche dans des réseaux de plus grande taille.

5.2.4 Méthode de validation proposée

Pour faciliter le déploiement d'une application de coopération véhiculaire, la validation poursuit essentiellement deux objectifs :

— Démontrer que l'algorithme aura le comportement désiré dans un scénario réel ;

— Déterminer quelles conditions suffisent à assurer son succès dans d'autres scénarios liés au même cas d'application.

Le premier objectif est assuré par le réalisme de la méthode de validation et de ses paramètres. Pour une démonstration formelle, il s'agira des hypothèses de travail, pour des tests routiers et émulés, des paramètres expérimentaux (dont le nombre de véhicules) et pour des simulations, de la qualité des modèles ainsi que des paramètres expérimentaux.

Type de validation	Scalabilité	Réalisme	Simplicité	Faible coût
Expérience routière	+	++++	+	+
Simulation	+++	+	+++	++
Émulation	++	++	++	+++
Preuve formelle	++++	+	+	++++
Méthode proposée	**++++**	**+++**	**++**	**++++**

TABLE 5.1 – Comparaison des méthodes de validation de la littérature et de la méthode proposée.

Le second objectif consiste à extrapoler les résultats à des situations similaires mais non identiques (par exemple changement d'emplacement d'une intersection ou d'une bretelle). Il n'est rempli que partiellement dans le cas de tests empiriques et uniquement à condition que les paramètres expérimentaux soient suffisamment représentatifs du type de mobilité attendu.

En plus de ces objectifs, les caractéristiques spécifiques de la méthode de prédiction sont à prendre en compte :

— le coût de mise en œuvre, une validation fréquemment utilisée à un niveau industriel se doit d'être à *bas coût* ;

— la *scalabilité*, qui représente la facilité de changement d'échelle, pour ajouter des véhicules au scénario de validation ;

— le *réalisme*, qui représente la conformité des résultats à ceux d'un scénario routier réel ;

— la *simplicité*, qui représente la facilité de mise en œuvre, à la fois d'un point de vue théorique (difficulté des calculs ou du travail d'analyse) mais aussi logistique (difficultés d'implémentation, de configuration du matériel ou de conduite des expériences).

Pour chaque méthode de validation (y compris la méthode proposée), une analyse qualitative de ses caractéristiques est proposée à la table 5.1.

La méthode proposée est conçue comme un intermédiaire entre l'analyse formelle et la validation empirique afin de répondre aux objectifs de validation. Elle dispose d'une partie des caractéristiques de validation empirique (un bon réalisme) et théorique (une meilleure scalabilité et un coût plus faible). Elle tire sa simplicité dans le fait de ne pas nécessiter une preuve complète (comme une validation empirique) ni une implémentation complète (comme une validation théorique). Il s'agit de prédire formellement le comportement d'un algorithme dans une série de mobilités donnée (elles doivent, dans l'idéal, être représentatives de la situation où le déploiement est envisagé). Elle se déroule en deux étapes :

— L'étude formelle de l'algorithme sous-jacent à l'application de coopération, visant à l'expression d'une propriété particulière, appelée propriété de performance topologique qui ne dépend que de la dynamique du réseau ;

— L'étude d'observations (réelles ou simulées) de réseaux dynamiques représentatives du cas d'usage d'intérêt, à la recherche de la propriété topologique de performance.

L'arrangement de ces étapes est décrit dans le schéma de la figure 5.1, où chaque étape est effectuée en parallèle jusqu'à la prédiction qui réunit les résultats des deux études.

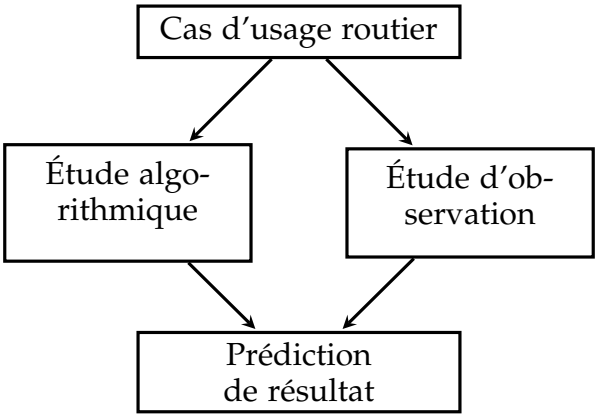


FIGURE 5.1 – Schéma du déroulement de la méthode de prédiction proposée

Alors que l'étude algorithmique peut s'avérer relativement complexe, elle se révèle généralement bien plus simple qu'une preuve complète, notamment grâce à l'utilisation du modèle des p-graphes dynamiques [**?**]. De plus, elle n'est pas renouvelée pour chaque observation de réseau dynamique mais n'est réalisée qu'une fois par cas d'application.

L'étude d'observations, au contraire, est plutôt simple, nettement plus qu'un test empirique, puisqu'aucune forme d'implémentation n'est nécessaire (contrairement aux simulations, émulations et expériences réelles). Elle est par ailleurs entièrement automatisable pour être réalisée sur de nombreux scénarios dynamiques différents. Les données de l'observation peuvent être collectées sur des tests routiers antérieurs, par un système de suivi automatisé de trafic (caméra de surveillance, détecteur de passage de véhicule, etc.). Leur analyse est automatisable par de petits programmes recherchant la propriété de performance topologique.

La méthode proposée repose ainsi sur une analyse théorique de l'algorithme et du cas d'usage appelée *étude algorithmique*. Grâce à cette analyse, il devient aisé d'étudier une observation de réseau dynamique pour prédire le comportement de l'algorithme dans cette observation. Cette dernière étape est appelée *étude d'observation*.

5.3 Étude algorithmique

Une étude formelle de l'algorithme d'une application de coopération véhiculaire dans son cas d'usage constitue la première étape de la prédiction. Cette étape vise à déterminer des conditions sur la topologie réseau, dans lesquelles le comportement de l'algorithme est prévisible.

5.3.1 Propriétés de performance algorithmique et topologique

L'objectif de l'étude formelle est de déterminer un prédicat nommé propriété de performance topologique, dépendant uniquement de la dynamique du réseau véhiculaire mais décrivant si les performances atteintes par l'algorithme sont satisfaisantes pour le cas d'usage, donc si l'utilisation de cet algorithme dans ce cas produit un succès.

La définition du succès d'un algorithme réparti découle directement de sa définition. Elle s'exprime sous forme de spécifications de l'algorithme qui pourraient par exemple être basées sur des propriétés de sûreté (*safety*) et de vivacité (*liveness*). Ces spécifications peuvent être regroupées sous la forme d'un prédicat décrivant le succès ou l'échec de l'algorithme à remplir ses spécifications.

Le cas d'application joue également un rôle dans le succès réel ou non d'une application. En effet, ses paramètres peuvent définir le type de données en jeu (et donc leur tailles), le nombre de véhicules impliqués ainsi que les paramètres techniques des communications (latence, débit, pertes) qui ont une grande influence sur l'issue de l'utilisation de l'algorithme.

La propriété de performance algorithmique est donc un prédicat, qui instancie les spécifications de l'algorithme au cas d'usage. Elle dépend à la fois des spécifications algorithmiques de l'application de coopération et des paramètres du cas d'usage. Elle décrit si les performances atteintes par l'algorithme réparti sont satisfaisantes au regard du cas d'application ou non.

L'étude algorithmique consiste ainsi à exprimer la propriété de performance topologique à partir de la propriété de performance algorithmique et d'une analyse de l'algorithme. Un schéma de la méthode détaillant un peu plus l'étude algorithmique sur la gauche est proposé à la figure 5.2.

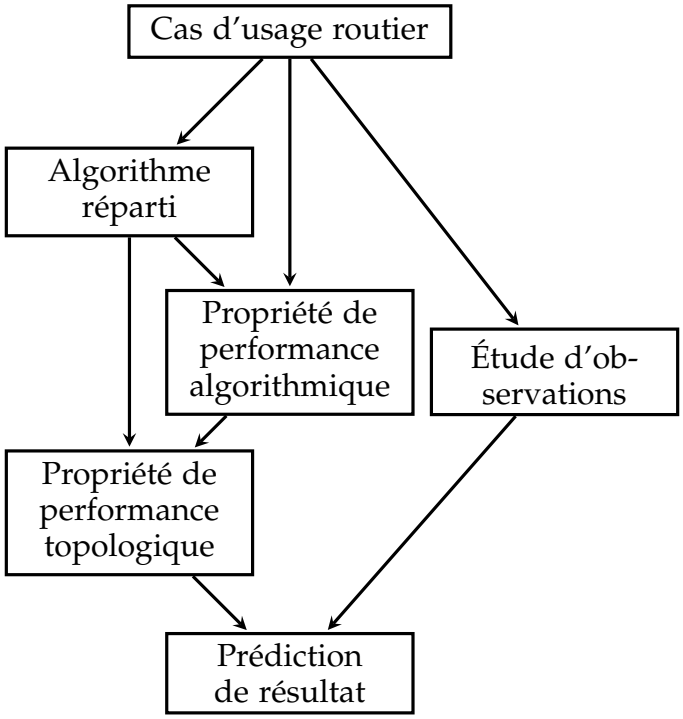


FIGURE 5.2 – Schéma du déroulement de la méthode de prédiction proposée détaillant l'étude algorithmique à gauche.

L'étude formelle vise à exprimer une propriété de performance topologique, dépendant uniquement de la dynamique du réseau véhiculaire, et reliée à la propriété prédite. Cette propriété est habituellement exprimée sous forme de conditions sur les p-graphes dynamiques représentant le scénario.

On peut analyser l'algorithme pour déterminer quelles séquences d'échanges de mes-

sages mènent à sa réussite. L'étude de ces séquences permet ensuite d'établir dans quelles types de p-graphes dynamiques elles peuvent avoir lieu ce qui résulte alors en une expression de la propriété de performances.

Puisque la propriété de performance topologique ne peut dépendre que de la dynamique du réseau, son expression résulte généralement d'hypothèses simplificatrices sur le fonctionnement de l'algorithme. La propriété de performance topologique n'est donc pas toujours (et n'est habituellement pas) strictement équivalente à la propriété prédite. Cette différence affecte le réalisme de la prédiction et il en résulte que les hypothèses simplificatrices doivent être choisies avec attention selon le résultat recherché.

5.3.2 Garanties attendues

L'utilisation de la méthode de prédiction proposée a notamment pour objectif de proposer des garanties de bon fonctionnement lors du déploiement d'une application de coopération véhiculaire.

Si la propriété de performance topologique est absolument équivalente à la propriété de performance algorithmique, une prédiction toujours exacte du résultat algorithmique peut être obtenue par l'analyse des p-graphes d'une observation. De cette manière, il est aisé de déterminer si l'application est adaptée à un contexte d'utilisation, si les paramètres techniques sont suffisants ou quelles performances maximales attendre.

Les hypothèses simplificatrices émises lors de l'étude formelle peuvent mettre à mal cette relation entre la propriété algorithmique et la propriété topologique. Il en résulte que la prédiction basée exclusivement sur l'analyse de la dynamique du réseau n'est pas parfaite et peut, dans certains cas, s'avérer inexacte.

En assurant que la propriété de performance est une propriété suffisante pour assurer la propriété prédite, mais pas forcément nécessaire, on garantit qu'une prédiction de succès algorithmique est toujours exacte, tandis qu'une prédiction d'échec algorithmique demeure incertaine. Ainsi, dans un contexte où l'application de coopération vise à améliorer la sécurité routière (fournir des informations sur des éléments non visibles, sur les vitesses pratiquées, etc.), prédire un succès algorithmique garantit une amélioration de la sécurité, ce qui a son importance pour choisir de lancer un déploiement. Si la prédiction est un échec algorithmique, cela ne signifie pas toujours que l'algorithme échouera en réalité, mais la prédiction demeurera prudente dans ce cas de figure. L'intérêt du déploiement est alors faible ou inexistant.

Naturellement, il est toujours possible de choisir des propriétés de performance extrêmement prudentes (il est même possible de choisir une propriété de performance impossible à réaliser ou presque), qui, si elles garantissent de ne pas illusionner un succès de l'algorithme, ne permettent jamais de prédire quelque chose d'utile (un succès algorithmique).

Le choix des hypothèses simplificatrices doit donc malgré tout, pour que la prédiction reste réaliste, limiter au maximum les différences entre propriété prédite et propriété de performance.

Lorsque la propriété de performance est nécessaire (mais pas forcément suffisante) pour la propriété prédite, une prédiction d'échec est assurée de correspondre à un échec réel. Cependant, une prédiction de succès demeure incertaine. Une telle situation ne permet pas d'offrir de garanties de sécurité, si l'application est impliquée dans la sécurité routière.

Si ce n'est pas le cas, la prédiction peut tout de même permettre de déterminer les performances maximales de l'application dans le contexte. Par ailleurs, si l'échec de l'application peut être détecté au cours de l'exécution, il peut également déclencher une réaction. La prédiction des cas d'échecs peut alors également permettre de prédire l'occurrence de ces réactions comme s'il s'agissait du succès d'un autre algorithme.

5.3.3 Exemples d'application de l'étude algorithmique

Considérons le cas d'usage dans lequel les véhicules arrivant sur une zone doivent recevoir une donnée trop massive pour être transmise en un seul message avant de pénétrer sur zone (mise à jour de carte détaillée, itinéraire de déviation).

Le gestionnaire du réseau routier installe avant la zone d'intérêt une borne de bord de route (*RSU*) disposant de la donnée à jour. La borne de bord de route exécute un algorithme réparti simple qui stipule qu'elle émet périodiquement (avec pour période de 1 s) et cycliquement un message contenant $\frac{1}{3}$ de la donnée fractionnée, comme le montre la figure 5.3. La borne émet en aveugle, elle n'a aucun retour sur la bonne réception de la donnée par les véhicules.

Dans ces conditions, l'algorithme est un succès si toute la donnée est reçue par chacun des véhicules approchant la borne au cours de l'exécution. La propriété de performance algorithmique est donc la bonne réception, par chacun des véhicules impliqués, de toute la donnée.

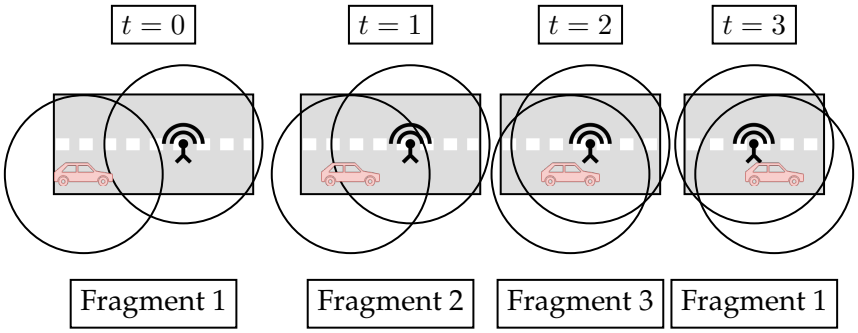


FIGURE 5.3 – Scénario d'exemple pour une prédiction. Les deux véhicules passent proches de la borne. Le véhicule rouge reçoit toute la donnée.

On peut facilement déterminer qu'un véhicule recevant 3 messages consécutifs obtiendra tous les fragments de la donnée même s'il manque la première émission du fragment 1. En effet, l'émission est cyclique et ordonnée, donc quel que soit le premier fragment de la série de 3 consécutifs, elle inclut les 3 différents fragments de donnée. L'existence d'un lien suffisamment long pour recevoir 3 messages consécutifs du RSU est donc suffisante pour assurer la propriété de performance algorithmique. On en déduit alors qu'il suffit que le l'observation comprenne un lien dont la durée est supérieure à $3 \times p$ reliant une voiture au RSU pour assurer que cette voiture reçoit toutes les données. On peut alors déterminer une propriété de performance topologique selon laquelle tous les véhicules doivent être reliés au RSU par au moins un lien de l'observation dont la durée est supérieure à $3 \times p$

Cette propriété topologique est suffisante à la réalisation de la propriété algorithmique mais pas nécessaire. En effet, un véhicule pourrait par exemple passer plusieurs

fois devant la borne avec une vitesse si grande qu'il ne puisse récupérer qu'un fragment à la fois. Après un certain nombre de passages (au moins 3), il disposerait de la donnée complète mais n'aurait jamais rempli la propriété de performance topologique déterminée ci-avant. Si une prédiction basée sur cette propriété de performance topologique résulte systématiquement en un succès, pour les scénarios représentatifs du cas d'usage, on peut en inférer la garantie que tous les véhicules du cas réel recevront la totalité de la donnée (si tel n'était pas le cas, c'est que le trafic utilisé n'était pas représentatif).

Lorsqu'il est compliqué d'obtenir l'expression de la propriété de performance directement en fonction de l'observation, il est parfois plus simple de l'exprimer en fonction des conditions sur la famille de p-graphes dynamiques représentant l'observation. En l'occurrence, il s'agirait simplement de l'existence pour chaque véhicule v d'une arête dans le 3-graphe dynamique (donc d'une 3-arête) reliant le véhicule v et la borne b.

En prenant une condition nécessaire mais pas suffisante, comme par exemple celle selon laquelle la somme des durées des liens entre la borne et chacun des véhicules est supérieure à $3 \times p$, on obtiendrait une prédiction capable de donner les performances maximales. En effet, si $15\,\%$ des prédictions sont des succès, cela signifie qu'au mieux $15\,\%$ des véhicules recevront toute la donnée dans le déploiement réel. Ce type de prédiction n'est pas destiné au transfert de données critiques pour la sécurité routière, mais plutôt à des applications de gestion du trafic, de la consommation de carburant ou de divertissement.

Il est également possible de choisir une propriété de performance qui n'est ni nécessaire ni suffisante pour assurer la propriété prédite, ce qui en fait seulement une approximation. La qualité de l'approximation dépend de la situation et ne permet pas d'inférer de garanties de sécurité routière.

La plupart des algorithmes destinés à un usage dans les réseaux véhiculaires prennent en compte le risque de perte non topologique aléatoires (liées à l'environnement radio, par exemple). Ces pertes peuvent survenir à tout moment au sein d'un lien de l'observation. On peut leur attribuer une probabilité basée sur des mesures en conditions réelles ou des modèles de propagation des signaux. La propriété de performance algorithmique doit tenir compte des pertes non topologiques que l'algorithme doit pouvoir supporter. On peut par exemple utiliser la probabilité de perte (taux de pertes) et l'intégrer au choix de la propriété de performance algorithmique.

Par exemple, on peut prendre un algorithme d'envoi périodique (de période $1\,\mathrm{s}$) d'envoi de balises (pour détection des véhicules voisins). Comme on peut le remarquer sur la figure 5.4, l'environnement radio provoque une probabilité de perte de 50 % pour chaque messages, on peut déterminer qu'un véhicule qui reste à portée d'un autre pendant une durée $2\,\mathrm{s}$ a une forte probabilité (75% si on compte une probabilité individuelle indépendante de 50% pour chaque message) d'en recevoir au moins 1. On pourrait alors prendre comme propriété de performance algorithmique la réception possible, dans l'observation d'au moins 2 balises. Selon le degré de garantie souhaitée, la valeur peut être choisie librement plus élevée.

L'étude algorithmique est destinée à déterminer une propriété de performance topologique indépendante de l'exécution de l'algorithme. Cette propriété renseigne cependant sur le comportement de l'algorithme dans l'observation, et peut même prendre en compte les pertes non topologiques de messages. Elle peut notamment être exprimée sous forme de condition sur la famille de p-graphes représentant l'observation. L'étude d'observation se base sur la propriété de performance topologique pour effectuer la pré-

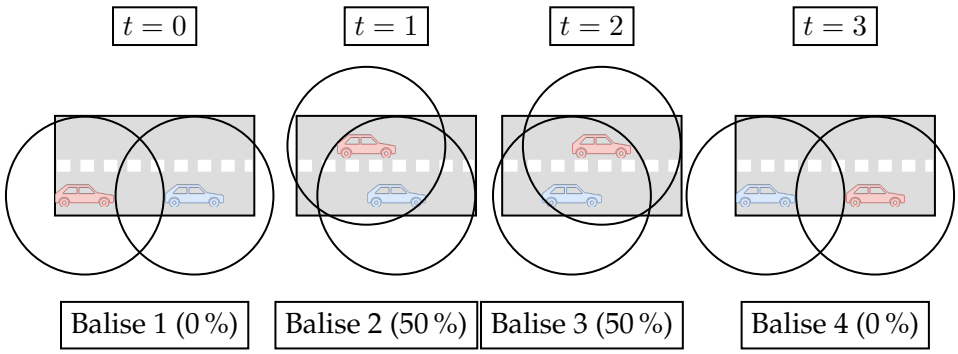


FIGURE 5.4 – Scénario d'exemple pour une prédiction dans un environnement avec pertes aléatoires de la moitié des messages. Les deux véhicules passent à portée l'un de l'autre. Si chaque message émis a une probabilité de 50 % de ne pas être reçu par un véhicule à portée de communication, chaque véhicule a 75 % de chances de recevoir au moins une balise de l'autre (25 % pour la 2, 25 % pour la 3, et 25 % pour les deux).

diction de résultat.

5.4 Prédiction par l'étude d'observation

Après avoir déterminé la propriété de performance découlant de l'application de coopération véhiculaire et du cas d'usage, l'analyse d'observation permet de prédire le comportement de l'algorithme. Elle se base sur la capture de graphes réalistes représentant des observations représentatives du cas d'usage, leur analyse permettant de déterminer un résultat prédit.

5.4.1 Capture de graphes réalistes

Une observation de réseau véhiculaire dans laquelle les liens seraient fiables est difficile à obtenir. En effet, une telle observation devrait tenir compte des toutes les pertes possibles, ce qui inclut les collisions de messages et interférences radio. Pire encore, cette observation serait alors beaucoup moins répétable, ce qui signifie que des véhicules reproduisant exactement le même scénario dynamiques produiraient une observation différente. Cette situation mettrait à mal la représentativité de l'observation, et donc l'utilité de la prédiction.

Il est tout à fait possible de capturer une observation au cours d'un test routier en conditions réelles. On peut pour cela par exemple équiper les véhicules expérimentaux d'un boîtier de communication par lequel on fait transiter des balises régulières comprenant l'identifiant du véhicule émetteur. Si chaque véhicule stocke sa position ainsi que les balises qu'il a reçues, il est possible de retracer les liens au cours du temps et, par conséquent, de produire l'observation. Avec cette approche, il faut toutefois prendre en compte que des pertes de paquets peuvent aléatoirement avoir lieu même si les véhicules restent en lien. On ne doit donc prendre en compte que les ruptures durables de communication pour former l'observation réelle.

Pour limiter les besoins logistiques de la capture de l'observation, le choix est fait de se baser plutôt sur les caractéristiques techniques du canal de communication utilisé. On peut par exemple utiliser la portée du système de communication utilisé pour associer

des véhicules réels mais non communiquants en fonction de leur position GPS. Cette position peut même être calculée depuis l'extérieur du véhicule, par exemple grâce à une caméra de surveillance (bien calibrée) sur la voie publique. De cette manière, il est possible d'obtenir de nombreuses observations d'une même zone selon diverses circonstances, et ainsi, d'assurer la meilleure représentativité possible du trafic.

Les positions GPS des véhicules peuvent également être virtuelles, par exemple lorsqu'elles sont constituées de traces issues de plusieurs tests ayant eu lieu à des moments différents (les véhicules impliqués n'ont donc jamais pu se rencontrer) ou de plusieurs fois la même trace décalée dans l'espace ou le temps. Il est même possible d'utiliser un simulateur de trafic comme SUMO [**?**] pour générer les traces GPS. Dans tous ces cas, on utilise ensuite la portée du moyen de communication afin de déterminer les liens ayant lieu au cours de l'observation. La figure 5.5 montre un exemple de capture d'observation à partir d'un scénario routier réel ou simulé, à partir de la portée du système de communication.

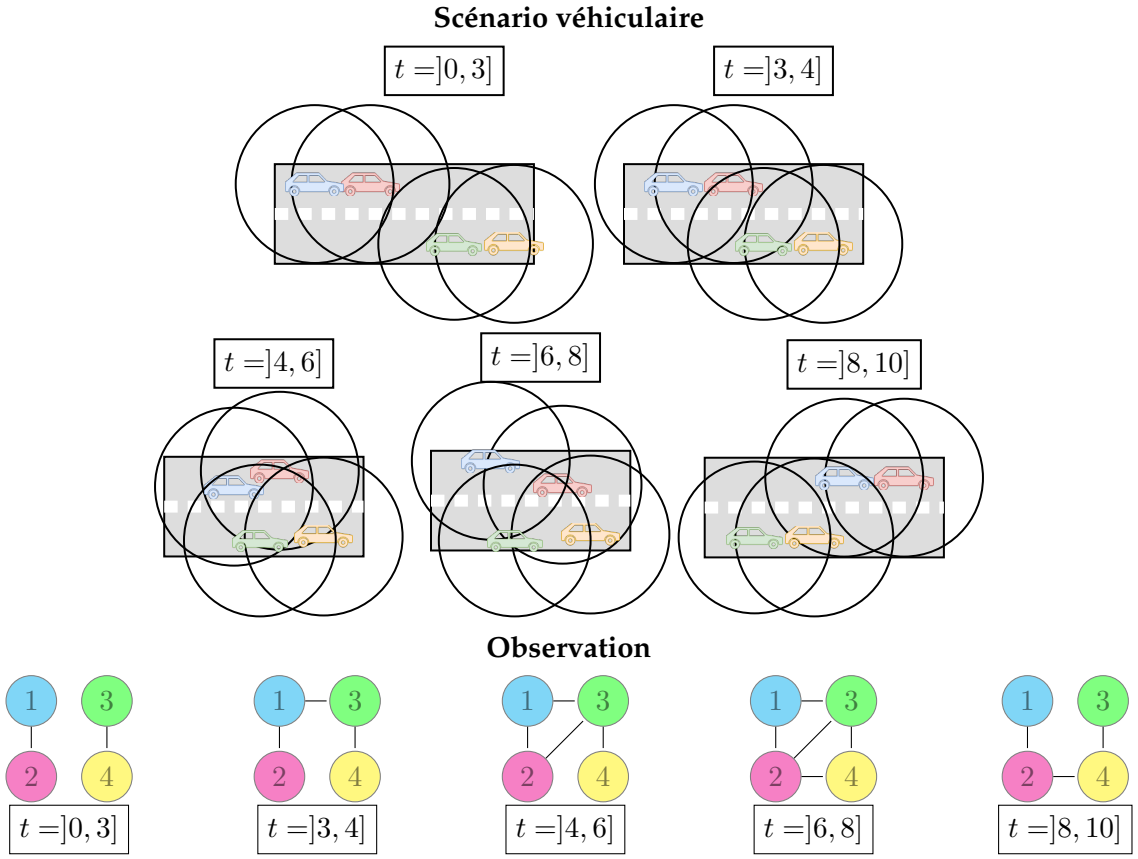


FIGURE 5.5 – Exemple de capture d'observation réaliste à partir d'un scénario véhiculaire.

L'utilisation de traces virtuelles ou simulées permet de réduire grandement les coûts d'obtention de l'observation par rapport à un test routier réel. Cela se fait au prix d'une petite perte de réalisme si les simulateurs de trafic sont performants. Cependant, il reste possible de récupérer les traces GPS de manière passive, avec des détecteurs de véhicules ou des caméras de surveillance (y compris grâce à des webcam publiques) par exemple, qui assurent un meilleur degré de réalisme.

5.4.2 Analyse de graphes

L'observation peut être utilisée pour calculer la famille de p-graphes qui la représente. En effet, cette modélisation est celle retenue pour exprimer la propriété de performance topologique. Construire la famille de p-graphes nécessite de connaître la fonction de durée de transfert $\delta : \mathbb{N} \to \mathbb{R}$ qui indique la durée nécessaire pour transférer un certain nombre de messages.

Dans le cadre des réseaux véhiculaires ad hoc, de nombreuses communications sont réalisées par l'envoi périodique de messages, dont la période d'envoi peut être fixe ou variable. C'est notamment le cas dans les standards de communication développés (CAM [**?**], BSM [**?**], CPM [**?**]). Cette fonctionnalité sert surtout à empêcher un nœud de surcharger le réseau en émettant en permanence. Lorsque cette période est fixée (à une valeur τ), on peut définir la fonction de durée de transfert comme suit : $\delta : p \to \tau \times p$. Lorsque la période τ est variable, on prend la valeur τ la plus grande que cette période peut prendre et l'expression de la fonction est inchangée.

Pour chaque entier p, on peut construire le p-graphe dynamique en analysant tous les liens de l'observation. Il suffit de ne retenir que ceux dont la durée est supérieure à $\delta(p)$, avant d'agréger les liens qui se produisent en même temps sur une durée supérieure à $\delta(p)$ sous forme de p-graphe, comme l'indique l'algorithme de construction proposé dans [**?**]. Ce procédé est décrit par la figure 5.6 à partir de l'observation présentée sur la figure 5.5.

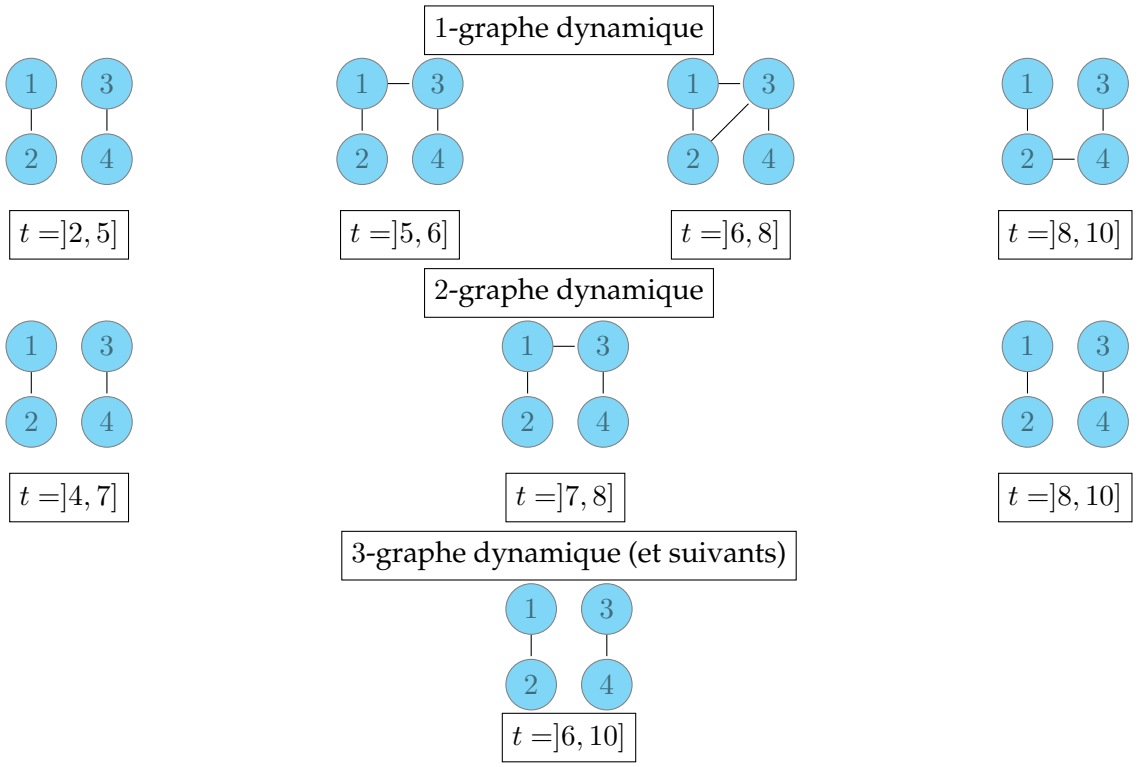


FIGURE 5.6 – Exemple de construction d'une famille de p-graphes $\mathcal{F}$ à partir de l'observation présentée en figure 5.5, en utilisant la fonction de durée de transfert simple $\delta(p) = 2 \times p$.

La famille représentant le scénario est ainsi construite avec un nombre fini de p-

graphes dynamiques (limité par la fonction δ et la durée de l'observation). Elle représente la topologie dynamique du réseau au cours de l'observation et peut être utilisée pour en étudier les propriétés.

5.4.3 Prédiction du résultat

La famille de graphes p-dynamiques représentant l'observation est utilisée pour y rechercher la propriété de performance topologique déterminée par l'étude algorithmique.

Lorsque la même propriété de performance topologique est recherchée dans de nombreuses observations, il est possible d'automatiser l'analyse de graphes (et notamment des p-graphes). Une telle automatisation permet de réaliser facilement de nombreuses prédictions afin de caractériser précisément un trafic aussi représentatif que possible de la situation réelle.

La présence de la propriété de performance topologique dans la famille de p-graphes dynamiques mène à une prédiction de succès algorithmique (on prédit que la propriété de performance algorithmique serait remplie par une exécution de l'algorithme dans cette observation). Son absence, en revanche, se solde par la prédiction d'un échec algorithmique (on prédit que la propriété de performance algorithmique ne serait pas remplie lors d'une exécution de l'algorithme dans cette observation).

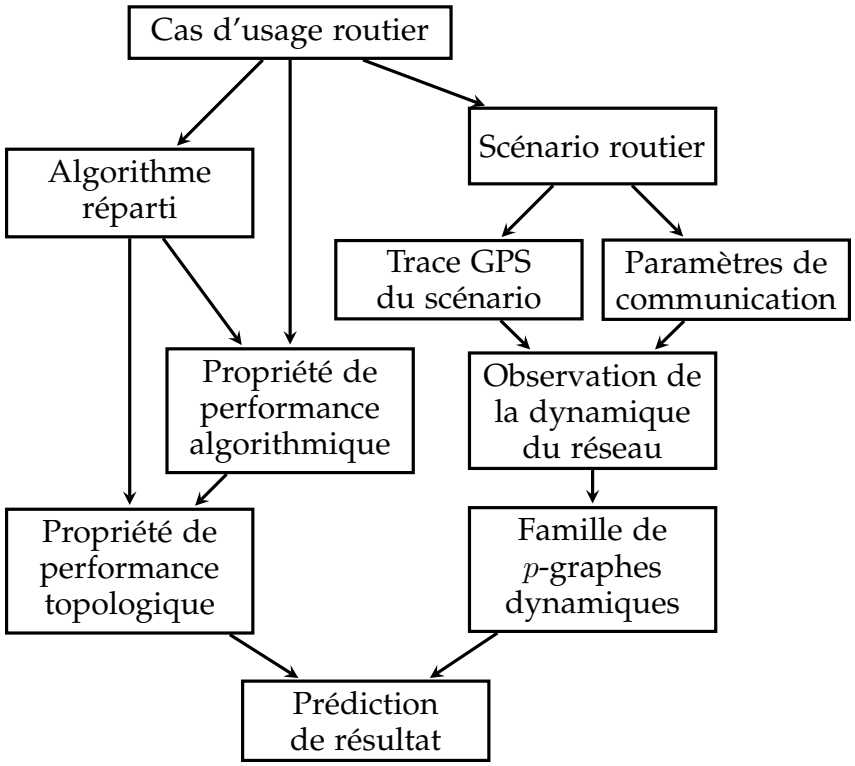


FIGURE 5.7 – Schéma du déroulement de la méthode de prédiction proposé détaillant l'étude d'observation à droite, et l'étude algorithmique à gauche.

La combinaison de l'étude algorithmique décrite en section 5.3 et de la prédiction permet la validation d'un algorithme de coopération véhiculaire dans un cas d'usage, comme le montre la figure 5.7. Ce schéma détaille le fonctionnement de toutes les étapes

de la méthode proposée, avec sur la gauche l'étude algorithmique et sur la droite, l'étude d'observation. La validation ainsi obtenue est à la fois à même de fournir des garanties de fonctionnement qui peuvent se traduire par des garanties de sécurité routière, mais aussi des prédictions de performances.

Les propriétés de performance algorithmique et topologique ne sont généralement pas exactement équivalentes. En effet, la perte d'information liée à la modélisation de la topologie dynamique et certaines hypothèses faites pour simplifier l'étude algorithmique empêchent parfois une équivalence parfaite entre ces deux propriétés. Dans ce cas, il est possible que la prédiction soit erronée. Lors de l'application de la méthode à des études de sécurité routière, il est primordial de savoir anticiper et comprendre l'impact de ces éventuelles erreurs de prédiction.

5.5 Conclusion

L'analyse d'un algorithme de coopération véhiculaire et la modélisation de la topologie réseau d'un scénario dynamique réel permettent de prédire le fonctionnement de cet algorithme dans le scénario modélisé. Cette prédiction est à même d'offrir des garanties attendues par les exploitants de réseaux routiers sur l'effet sur la sécurité routière du déploiement applicatif. L'étude d'observations peut être automatisée et être ainsi menée sur un grand nombre d'observations, assurant une bonne représentativité de l'étude menée. Elle est compatible avec tous les modèles de graphes dynamiques, mais le choix du modèle utilisé pour représenter le scénario peut avoir un impact sur les propriétés qui peuvent être étudiées, donc sur les scénarios et algorithmes utilisables.

Une mise à l'épreuve expérimentale de la méthode de prédiction peut permettre de démontrer à la fois sa faisabilité et de vérifier son pouvoir prédictif. Elle permettra également d'évaluer la difficulté de l'étude formelle qui ne semble, en revanche, pas facilement automatisable.

Chapitre 6

Évaluation empirique de la méthode de prédiction proposée

Sommaire

6.1 Introduction

Pour le développement comme pour le déploiement d'applications de coopération véhiculaires, il est primordial de valider la solution envisagée. Cette validation est généralement réalisée par plusieurs méthodes complémentaires.

Une méthode développée dans le cadre de ce travail est proposée dans [**?**, **?**], afin de permettre la validation d'une application de coopération véhiculaire par une étude de l'algorithme sous-jacent (étude algorithmique) assortie d'une étude d'observations, qui consiste à rechercher certaines propriétés dans des topologies dynamiques de réseaux véhiculaires. Cette méthode présente l'intérêt de ne pas nécessiter d'expérimentations (pas d'implémentation) ni de train logistique complexe (pas de matériel) tout en assurant certaines propriétés (ayant un impact sur la sécurité routière) dans un contexte réaliste.

Cette méthode est cependant sujette à des erreurs de prédictions liées aux hypothèses effectuées lors de l'étude algorithmique et à la modélisation de la topologie réseau dynamique. Une évaluation expérimentale du pouvoir prédictif de la méthode est donc nécessaire, elle est notamment conduite dans [**?**].

On établira tout d'abord un protocole expérimental à même d'analyser le pouvoir prédictif de la méthode proposée, avant de présenter les résultats et leurs interprétations. Une discussion sur les résultats et perspectives est enfin conduite.

6.2 Protocole expérimental

L'évaluation de la méthode de prédiction proposée au chapitre 5 s'effectue par des expériences en émulation. Elle consistera à comparer les prédictions obtenues par la méthode au résultat d'une exécution réelle dans les mêmes conditions. Ce résultat d'exécution est appelé vérité terrain.

6.2.1 Vérité terrain

Pour chaque algorithme étudié, une implémentation sous forme d'application répartie est réalisée. Elle s'effectue via la suite logicielle Airplug, développée au laboratoire et qui permet le développement rapide d'applications réparties en Python, le langage utilisé dans le cadre de ce travail, ou dans d'autres langages (C, tcl, etc.) comme cela est décrit au chapitre 2. Ces applications peuvent ensuite être à la fois utilisées en émulateur et en expériences réelles, sans modifications supplémentaires.

Chaque application est à même de stocker dans des fichiers de log les messages qu'elle reçoit et ceux qu'elle envoie, ainsi que les résultats de ses calculs internes. Elle peut choisir de communiquer avec d'autres applications locales (s'exécutant sur le même nœud) ou des applications distantes.

La suite logicielle Airplug propose un émulateur capable d'exécuter une application à l'identique d'une expérimentation réelle, tout en émulant le réseau de communication grâce à un système structuré autour de pipes linux. L'émulateur permet de gérer la trajectoire des véhicules et peut être paramétré avec des données issues d'expériences réelles (traces GPS, taux de pertes, portée du signal, latence, etc.) grâce auxquelles il permet d'effectuer des expériences réalistes [**?**].

La *vérité terrain* représente le résultat réel de l'application dans le même réseau véhiculaire que celui sur lequel la prédiction a lieu. L'émulateur permet d'exécuter l'application répartie dans le réseau émulé, ce qui donne accès, par une analyse des fichiers de log de l'application, au comportement de l'application durant l'exécution (messages échangés, calculs effectués) dans ce réseau.

Sur chaque véhicule, l'émulateur exécute l'application répartie et assure les communications entre véhicules. Après l'exécution du scénario, il est possible de lire les fichiers de log pour déterminer le comportement de l'application. Analyser les fichiers de log permet d'y rechercher la propriété de performance algorithmique, qui décrit le fonctionnement satisfaisant de l'algorithme.

Par exemple, dans le scénario décrit en section 5.3, il serait possible d'étudier, pour chaque véhicule, quel était le contenu des messages qu'il a reçus. En s'assurant qu'ils contiennent la totalité de la donnée (les 3 fragments), il est possible de conclure que la vérité terrain de cet essai est un succès. Si un ou plusieurs fragments manquent, on peut

conclure que la vérité terrain sur cet essai est un échec. Cette analyse est conduite via de petits scripts utilisant notamment des expressions régulières destinées à vérifier la propriété de performance algorithmique.

6.2.2 Capture des p-graphes

Lors de chaque expérience émulée, l'émulateur a besoin de connaître la topologie réseau. En effet, c'est lui qui assure la transmission des messages d'un véhicule à l'autre et d'une application à l'autre. Il calcule pour cela en permanence à partir des paramètres du protocole de communication (portée) et des positions géographiques des nœuds (traces GPS) les liens de communication entre les nœuds. Il peut donc sauvegarder lui aussi, dans un fichier de log, la topologie du réseau au cours du temps. Ce fichier de log contient alors une description complète de l'observation du réseau au cours de l'expérience.

Cette observation peut être analysée pour en déduire la famille de p-graphes dynamiques représentant le scénario, Cette analyse est réalisée via un programme nommé *pgraphtool*, écrit en C++ au laboratoire et déjà utilisé par le passé. Il est capable d'analyser directement le fichier de log de l'émulateur pour calculer la famille correspondante (jusqu'à une valeur limite p_{max} de p). Sa sortie est la famille de p-graphes dynamiques représentant le scénario, et elle peut également être analysée grâce à des scripts comprenant des expressions régulières.

L'analyse de propriétés des graphes peut être assez fastidieuse (complexité importante), ce qui explique que les scripts d'analyse de la topologie sont parfois assez lourds. Elle consiste à vérifier la propriété de performance topologique, et n'a pour seules données d'entrée que le fichier de log de l'émulateur représentant l'observation du réseau dynamique au cours du temps. Si la propriété de performance topologique est réalisée par la famille de p-graphes dynamiques, un succès est prédit, tandis que si elle n'est pas réalisée, un échec algorithmique est prédit.

6.2.3 Qualité de prédiction

La comparaison entre la prédiction et la vérité terrain donne la*qualité de prédiction*. Cette dernière peut prendre 4 valeurs différentes :

— Succès prédit : la propriété de performance topologique est remplie par l'observation et la propriété de performance algorithmique est remplie par l'exécution (un succès a été prédit et la vérité terrain est un succès) ;

— Échec prédit : la propriété de performance topologique n'est pas remplie par l'observation et la propriété de performance algorithmique n'est pas remplie non plus par l'exécution (un échec a été prédit et la vérité terrain est un échec) ;

— Succès non prédit : la propriété de performance topologique n'est pas remplie par l'observation mais la propriété de performance algorithmique est remplie par l'exécution (un échec a été prédit mais la vérité terrain est un succès) ;

— Échec non prédit : la propriété de performance topologique est remplie par l'observation mais la propriété de performance algorithmique n'est pas remplie par l'exécution (un succès a été prédit mais la vérité terrain est un échec).

Puisque les expériences visent à mettre la méthode à l'épreuve, les scénarios d'étude doivent être définis de manière à mettre à l'épreuve la capacités prédictive de la méthode

proposée. Ainsi, la propriété de performance algorithmique n'est pas choisie de manière à répondre, dans les scénarios étudiés, à un impératif applicatif dicté par le cas d'usage, mais plutôt à un besoin de mettre la méthode proposée en difficulté.

En effet, il est toujours possible de proposer une propriété de performance topologique très prudente (ou éventuellement très optimiste), qui consisterait par exemple à prédire systématiquement un échec (ou éventuellement systématiquement un succès). Une telle propriété de performance topologique serait effectivement suffisante (ou nécessaire) pour réaliser la propriété de performance algorithmique, quelle qu'elle soit. Pour pouvoir s'assurer que la propriété de performance topologique ait une valeur prédictive, il faut donc au minimum qu'elle prédise des succès et des échecs dans chaque expérience. Par ailleurs, pour assurer que les résultats soient significatifs, les succès algorithmiques comme les échecs algorithmiques doivent être suffisamment représentés. En effet, un scénario trop facile ou trop difficile assurerait une domination énorme des succès (prédits ou non), ou respectivement des échecs, rendant l'analyse de la proportions d'échecs prédits (ou respectivement de succès prédits) au sein des échecs (ou respectivement des succès) plus hasardeuse.

Ainsi, les propriétés de performance algorithmiques choisies dans le cadre de cette thèse ont pour objectif de permettre les deux issues applicatives dans des proportions significatives, afin de pouvoir explorer les proportions de prédictions exactes dans les deux issues.

La qualité des résultats de prédiction sera représentée sous forme de graphiques tels que présentés sur la figure 6.1. Pour chaque scénario, est représentée horizontalement la proportion de succès et d'échecs (la vérité terrain), tandis que l'ordonnée représente la proportion de prédictions exactes au sein de chaque catégorie.

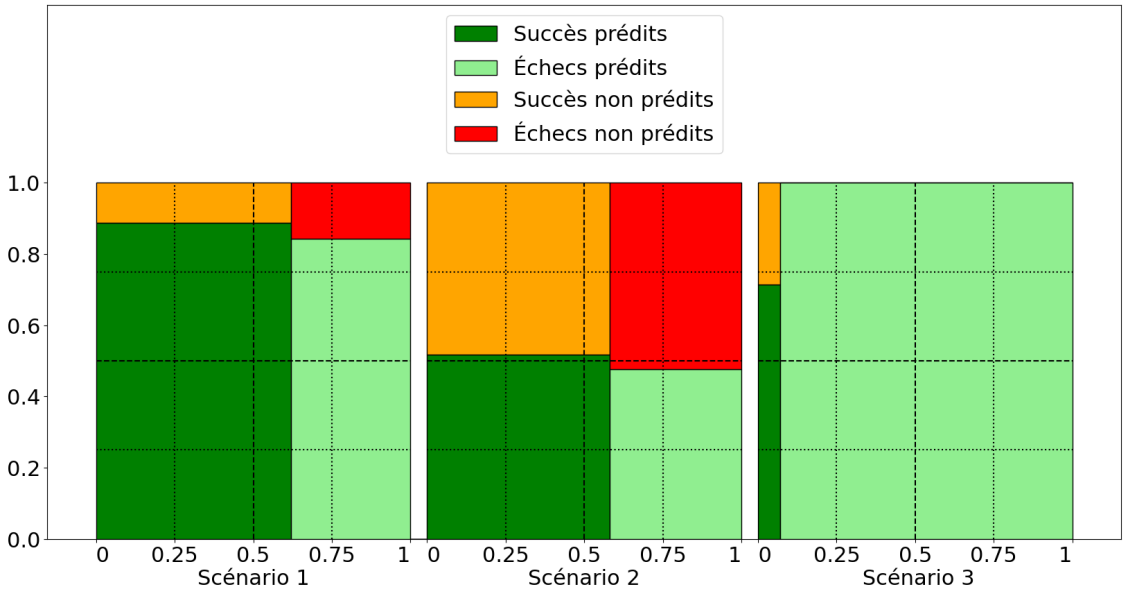


FIGURE 6.1 – Exemple fictif de graphique représentant la qualité de prédiction sur 3 scénarios différents.

Ainsi, dans le scénario 1, 62 % des tests se sont soldés par des succès applicatifs contre 38 % d'échecs applicatifs (vérité terrain). Parmi les succès, 90 % furent correctement prédits (soit 55 % des tests qui se soldent par un succès prédit) et parmi les échecs, 85 %

l'ont été (soit 32 % des tests qui se sont soldés par des échecs prédits). Il s'agit du résultat qu'on attend d'une expérience où la propriété de performance topologique est une bonne approximation de la propriété de performance algorithmique sans lui être nécessaire (présence de succès non prédits) ni suffisante (présence d'échecs non prédits).

Le scénario 2, quant à lui, montre près de 60 % de succès applicatifs, dont tout juste la moitié (soit 30 % des tests) ont été correctement prédits. Les échecs représentent 40 % des tests, dont la moitié (soit 20% des tests environ) ont été correctement prédits. Ce résultat montrerait que la propriété de performance topologique est une mauvaise approximation de la propriété de performance algorithmique, ni nécessaire ni suffisante, et qui donne une idée trompeuse des performances possibles.

Le scénario 3 montre que 93 % des tests se soldent par un échec applicatif, mais que ceux-ci sont toujours correctement prédits. Cette absence totale d'échecs non prédits, est attendue lorsque la propriété de performance topologique est suffisante à assurer la propriété de performance algorithmique. Parmi les succès algorithmiques, 75 % ont été correctement prédits. La propriété de performance algorithmique était manifestement trop difficile à réaliser, puisque les succès applicatifs sont largement minoritaires. Du fait de la proportion faible de succès dans par rapport à tous les essais, la proportion de succès prédits n'est pas forcément représentative de la réalité et des tests dans des scénarios similaires mais plus favorables à la propriété de performance algorithmique devraient être conduits.

Lorsqu'une propriété de performance topologique suffisante à la propriété de performance algorithmique est utilisée, ce qui se matérialise expérimentalement par une absence totale d'échecs non prédits (la zone rouge sur le graphique), il devient possible, lorsque la prédiction a lieu sur un trafic représentatif du cas d'usage et ne produit que des succès, d'en déduire que le déploiement réel de l'application fonctionnera systématiquement et, s'il est critique pour la sécurité routière, qu'elle sera préservée. Les graphiques ressemblant au scénario 3 mais comprenant plus de succès applicatifs sont ceux correspondant aux situations où il est possible de valider un déploiement grâce à la méthode proposée.

La proportion de succès non prédits dans au cours des tests d'évaluation de la méthode proposée n'est pas forcément importante, si la proportion de succès prédits représente une fraction significative des succès applicatifs. En effet, la validation n'est acceptable pour le gestionnaire d'un réseau que dans les cas où l'immense majorité des prédictions donnent un succès (la proportion d'échecs tolérés représente le risque qu'accepte de courir le gestionnaire du réseau routier). Ainsi, les succès non prédits n'affectent que la quantité de scénarios exclus de la validation par la méthode proposée.

6.3 Résultats

Le protocole expérimental a été mis en œuvre sur différents cas d'usage, afin notamment de couvrir des scénarios et des problèmes algorithmiques de natures différentes. Tout d'abord, un simple algorithme d'envoi périodique de balises sera utilisé. Il permettra de valider le protocole expérimental en réalisant des tests à la fois sur des scénarios autoroutiers et urbains. Ensuite, un algorithme de transmission de données avec acquittement fera l'objet d'expériences, avant un algorithme de diffusion. Enfin, un algorithme de découverte de voisinage puis sa variante servant à découvrir les triangles topologiques seront analysés par cette méthode.

6.3.1 Beaconing autoroutier

Afin de valider le protocole expérimental et les outils de son implémentation, un premier algorithme simple est étudié. Cet algorithme est nommé *Beacon* et est décrit dans l'algorithme 4. Il s'agit d'un algorithme qui envoie périodiquement des messages d'identification appelés balises (ou *beacon*). Ce type d'algorithme a des applications importantes en réseaux véhiculaires, notamment car il est à la base de protocoles standard comme l'envoi de messages CAM [**?**] ou BSM [**?**].

Algorithme 4 : **Beacon sur le nœud *id***

1 Initialisation :
  ▷ *Numéro de séquence de la balise*
2 $nseq \leftarrow 0$
3 Expiration du timer :
4 **envoyer**(source=*id*, seq=$nseq$)
5 $nseq \leftarrow nseq + 1$
6 Réarmer le timer
7 Réception d'une balise :
8 **recevoir**(source=s, seq=n)
9 Ajouter (s,n) au journal (log)

Le scénario utilisé se déroule sur l'autoroute A29 entre Amiens et Saint Quentin. Sur cette autoroute, un camion circule à 90 km/h lorsqu'une voiture placée derrière lui le dépasse en raison d'une vitesse supérieure (110,120 ou 130 km/h selon les tests). Un schéma des trajectoires du scénario est proposé à la figure 6.2. Le camion comme la voiture démarrent aléatoirement sur leur trajectoire, la voiture partant toujours derrière le camion.

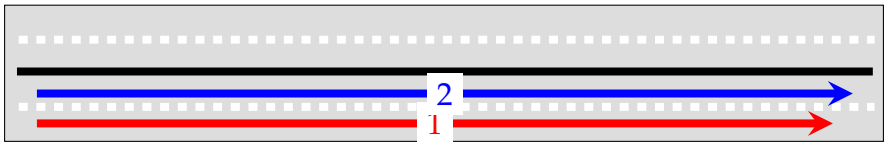


FIGURE 6.2 – Schéma décrivant les trajectoires des véhicules dans le scénario de dépassement sur autoroute. La trajectoire 1 représente le camion et la trajectoire 2 la voiture qui le dépasse.

On choisit de considérer qu'un échec de l'application a eu lieu pour tous les tests ne permettant pas à la voiture d'obtenir au moins 20 beacons différents du camion. Cette valeur est choisie arbitrairement pour permettre aux tests d'avoir une issue incertaine. Avec l'algorithme beacon, pour recevoir 20 messages différents du camion, il faut que la capacités totale des 1-arêtes reliant le camion et la voiture soit supérieure ou égale à 20, comme l'indique l'équation 6.1. En effet, l'existence d'une 1-arête garantit la réception d'une balise tandis que l'existence d'une 1-arête e dont la capacité est $C(e) = p$ assure la réception de p messages de balises successifs.

$$PP_{\text{Beacon-autoroute}} \equiv \sum_{\substack{e\,:\,1-arete \\ (camion,\,voiture)}} C(e) \geq 20 \tag{6.1}$$

La qualité des prédictions réalisées au cours de ces expériences est représentée sur la figure 6.3, selon la vitesse de la voiture dans le scénario. La proportion de succès applicatifs est de 72 % à 110km/h, 55 % à 120 km/h et 90 % à 130km/h. Dans tous les scénarios, la totalité des prédiction se sont avérées conformes à la vérité terrain, et ce quelle que soit l'issue de l'algorithme (succès ou échec applicatif) ce qui valide l'implémentation de la méthode et sa faisabilité dans des situations réalistes plus complexes.

Compte tenu de la nature simple de l'algorithme (une instances locale ne réagit pas réellement à la réception de données d'une autre) et du scénario (impliquant deux nœuds, dont la vitesse est fixe), d'autres méthodes de validation auraient pu s'avérer suffisantes (simulations ou émulations, notamment). Cependant, la méthode proposée pourrait tout de même être utile car elle peut être utilisée avant même que l'algorithme soit implémenté en application. Ainsi, les efforts de développements ne seraient entrepris qu'après avoir vérifié que l'application atteindrait des performances suffisantes dans le cas envisagé.

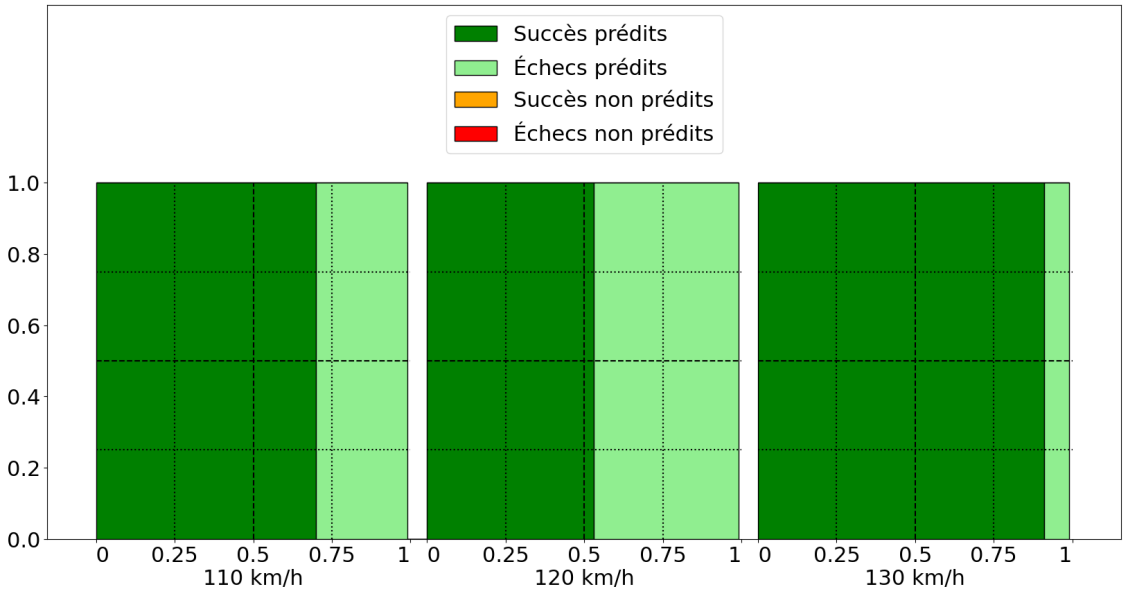


FIGURE 6.3 – Qualité des prédictions pour les expériences impliquant l'algorithme Beacon dans le scénario d'un dépassement autoroutier à différentes vitesses

6.3.2 Beaconing urbain

La difficulté des expériences est alors augmentée d'un cran pour prendre en compte des scénarios plus complexes, toujours avec l'algorithme Beacon. Ces scénarios se déroulent en contexte urbain et prendront en compte les propriétés particulières de la circulation en ville (intersections plus fréquentes, vitesses plus basses, trajectoires plus tortueuses, arrêts possibles, etc.).

Le premier scénario est centré autour d'un feu tricolore. Dans ce scénario, trois véhicules tournent à droite sur une intersection équipée d'un feu tricolore. Leur feu est vert, ils passent sans s'arrêter. Venant de leur droite, un véhicule souhaite, quant à lui, tourner à gauche sur la même intersection, mais rencontre un feu rouge qui l'oblige à ralentir, puis s'arrêter.

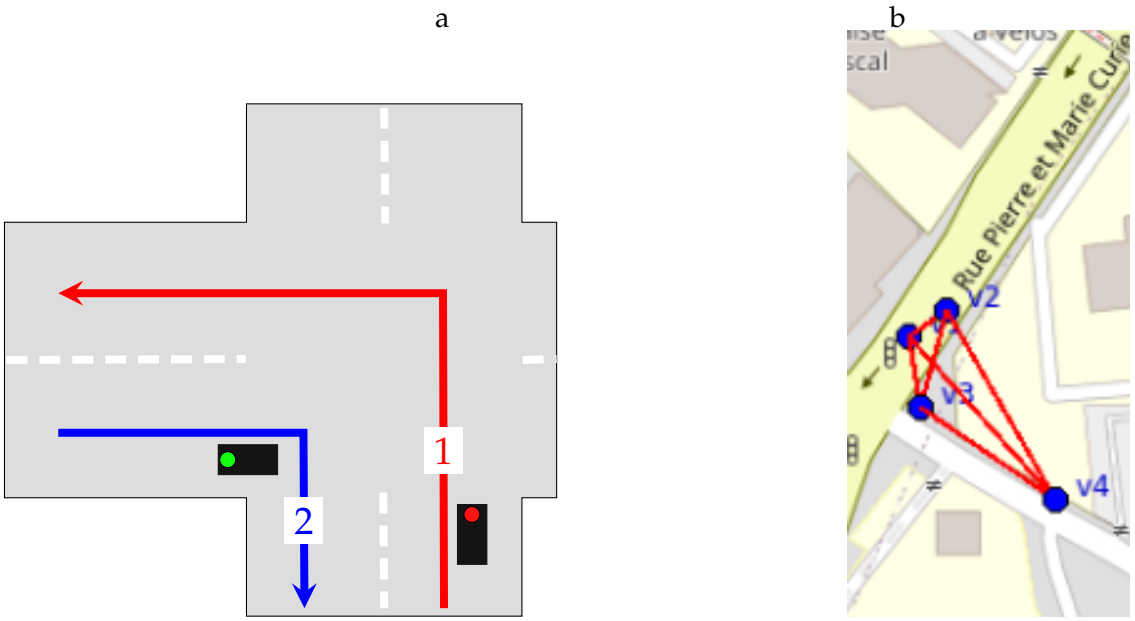


FIGURE 6.4 – Description du scénario urbain centré sur un feu tricolore. Sur le schéma de la figure a, la trajectoire 1 représente le véhicule d'intérêt qui a le feu rouge et la trajectoire 2 le convoi de véhicules qui ont le feu vert. Sur la capture d'écran de la figure b, l'émulateur exécute le scénario et les liens rouges représentent les communications disponibles.

Il démarre aléatoirement sur sa trajectoire avant l'intersection. Une illustration de ce scénario est proposée en figure 6.4 à travers un schéma et une capture d'écran de l'émulateur au cours d'une exécution.

Dans ce scénario, la propriété de performance algorithmique choisie est que le véhicule qui s'arrête au feu rouge reçoit au moins 15 balises de chacun des trois véhicules du convoi qu'il croise. On en déduit la propriété de performance topologique présentée à l'équation 6.2 selon laquelle pour chaque véhicule i du convoi en sens inverse, la somme de capacités des 1-arêtes le reliant au véhicule rencontrant le feu rouge doit être supérieure à 15.

$$PP_{\textit{Beacon-feu-tricolore}} \equiv \quad \forall i \in \{1,2,3\}, \sum_{\substack{e\,:\,1-arete \\ (v_1,v_i)}} C(e) \;\geq\; 15 \tag{6.2}$$

La qualité des prédictions réalisées au cours de ces expériences est représentée sur la figure 6.7 à gauche, sous le nom " Feu tricolore ".

Un scénario centré sur une voie rapide urbaine de Compiègne encadrée par deux ronds points est également étudié. Sur cette voie rapide sont placés aléatoirement 6 véhicules qui la parcourent jusqu'au rond point sur lequel ils font demi-tour.

Dans ces conditions, certains véhicules se croisent sur la voie rapide et d'autres s'approchent suffisamment uniquement lorsqu'ils ralentissent pour s'insérer sur le rond-point. Une illustration de ce scénario est proposée grâce à une capture d'écran réalisée sur l'émulateur à la figure 6.5.

La propriété de performance algorithmique choisie sera alors pour le véhicule d'intérêt v_1 d'avoir reçu 5 balises pour au moins 3 véhicules différents. L'expression de la

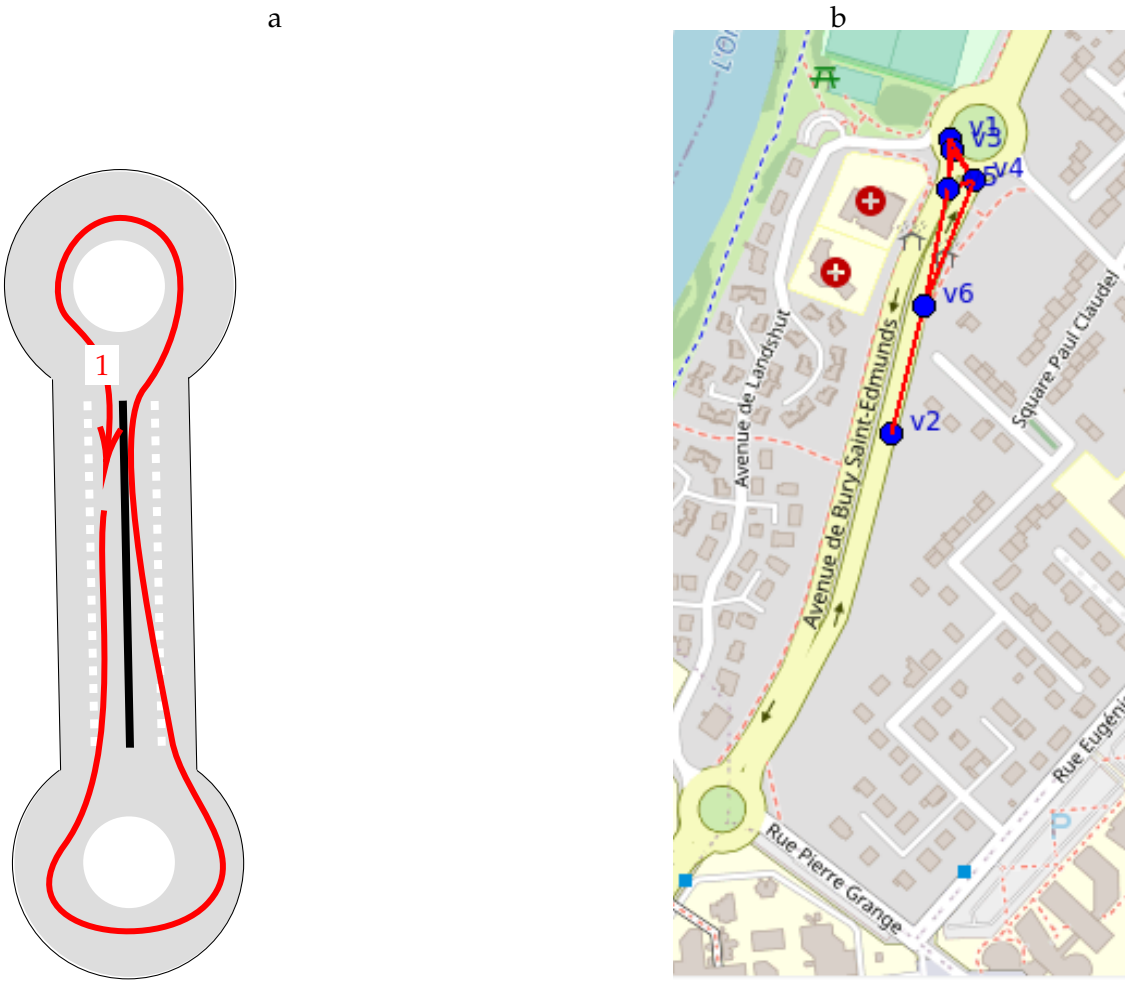


FIGURE 6.5 – Description du scénario urbain centré sur les ronds-points d'une voie rapide urbaine. Sur le schéma de la figure a, la trajectoire 1 représente celle de tous les véhicules du scénario. Sur la capture d'écran de la figure b, l'émulateur exécute le scénario et les liens rouges représentent les communications disponibles.

propriété de performance topologique est alors plus complexe, et fera appel à la fonction $f(v,i)$ qui associe à chaque véhicule v, 1 si au moins i balises émanant de lui ont été reçues, et 0 sinon, comme défini à l'équation 6.3.

$$\begin{aligned} f_{v_1}(v,i) = 1 \quad &\equiv \quad \sum_{\substack{e\,:\,1-arete \\ (v_1,v)}} C(e) \geq i \\ f_{v_1}(v,i) = 0 \quad &\equiv \quad \sum_{\substack{e\,:\,1-arete \\ (v_1,v)}} C(e) < i \end{aligned} \tag{6.3}$$

La propriété de performance topologique est alors que la somme des $f_{v_1}(v,5), \forall v$ soit supérieure à 3 comme décrit à l'équation 6.4.

$$PP_{\text{Beacon-rond-point}} \equiv \sum_{i \in \{2,\ldots,6\}} f(v_i,5) \;\geq\; 3 \tag{6.4}$$

La qualité des prédictions réalisées au cours de ces expériences est représentée sur la figure 6.7, sur le graphique du milieu, sous le nom " Rond point ".

Un scénario de trafic dense est également testé avec 48 véhicules évoluant sur 4 trajectoires dans la ville de Compiègne sur lesquelles ils démarrent aléatoirement et avec une vitesse prédéfinie de 30 ou 40 km/h. Une illustration de ce scénario est proposée en figure 6.6.

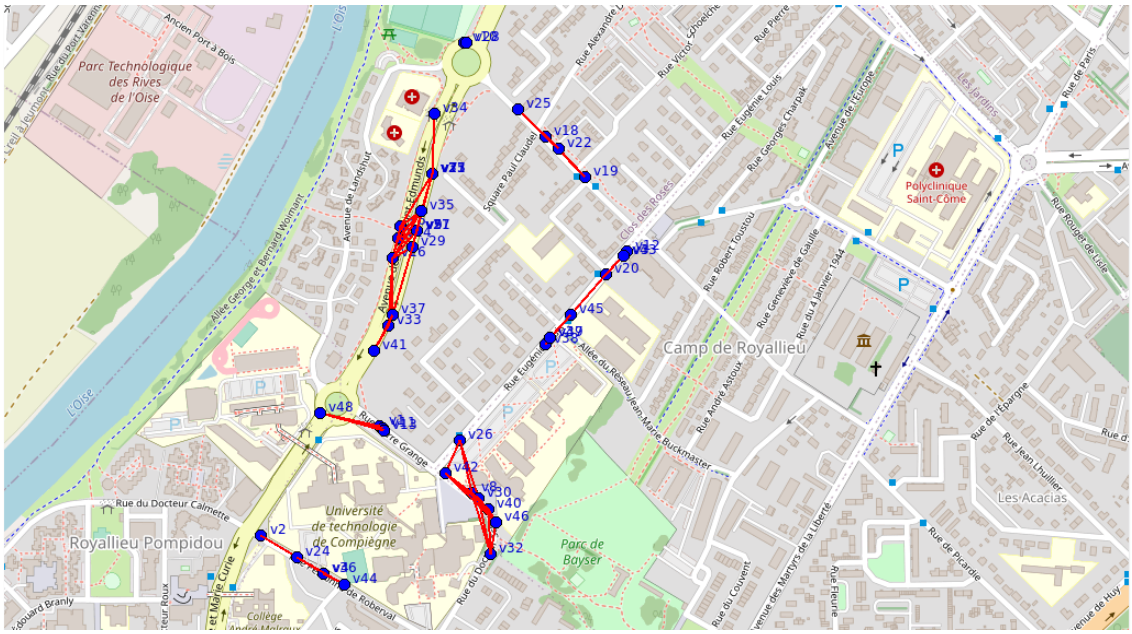


FIGURE 6.6 – Capture d'écran de l'émulateur exécutant le scénario urbain au trafic dense. Les liens rouges représentent les communications disponibles.

Dans ces conditions, la propriété de performance algorithmique choisie est, pour le véhicule d'intérêt v_1, d'avoir reçu au moins 5 balises, pour au moins 10 véhicules différents. Elle permet de déduire la propriété de performance topologique détaillée à l'équation 6.5 selon laquelle la somme des $f_{v_1}(v, 5), \forall v$ doit être supérieure à 10.

$$PP_{\text{Beacon-dense}} \equiv \sum_{i \in \{2,\ldots,48\}} f_{v_1}(v, 5) \quad \geq \quad 10 \tag{6.5}$$

La qualité des prédictions réalisées au cours de ces expériences est représentée sur la figure 6.7, sur le graphique de droite, sous le nom " Trafic dense ".

Dans le scénario « Feu tricolore », environ 28 % des tests se soldent par des succès, une proportion qui montre que les deux issues sont réellement possibles. Le scénario « Rond point » permet à 80 % des expériences d'assurer des succès applicatifs, les deux issues restent alors possibles dans ces expériences. Dans ces deux scénarios, la totalité des prédictions s'avèrent conformes à la vérité terrain (pas de succès non prédits ni d'échecs non prédits).

Dans le scénario « Trafic dense », un succès applicatif est observé dans environ 60 % des tests. Même si tous les échecs applicatifs ont été correctement prédits (pas d'échecs non prédits), des erreurs de prédictions apparaissent cependant concernant des succès algorithmiques non prédits. Ces situations rares (moins de 10 %) des expériences n'affec-

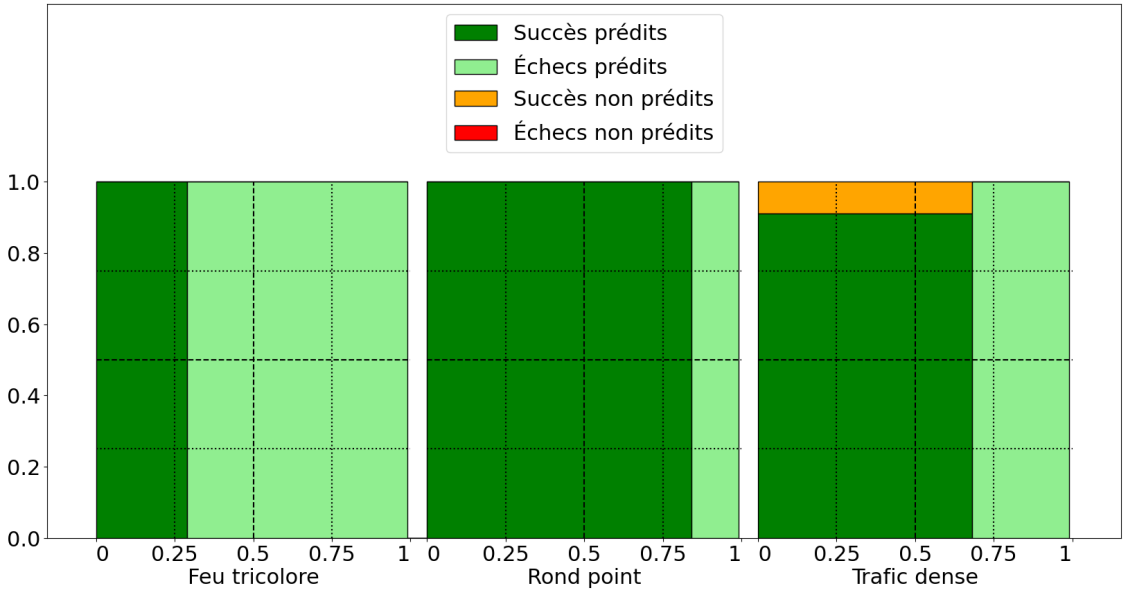


FIGURE 6.7 – Qualité des prédictions pour les expériences impliquant l'algorithme Beacon dans des scénarios urbains

teraient pas la sécurité routière si l'application était critique : elle aurait bien fonctionné même si cela n'avait pas été anticipé.

6.3.3 Transmission de donnée acquittée sur autoroute

Dans un cas d'usage plus complexe, un algorithme de transmission d'une donnée (comme par exemple une carte routière) par fragments est étudié. La source de la donnée transmet périodiquement un fragment et les nœuds receveurs à portée envoient des accusés de réception lorsqu'ils ont reçu $n (n \in \mathbb{N})$ fragments. Cet accusé de réception permet à la source d'ajuster le transfert pour viser les fragments manqués afin que le nœud receveur reçoive tous les fragments successifs de la donnée. Lorsqu'un receveur quitte le voisinage de la source de donnée, il commence à se comporter comme une source de données.

Cet algorithme est étudié dans des scénarios autoroutiers où la donnée fragmentée est détenue par une borne de bord de route (*RSU*) tandis que les nœuds receveurs sont un camion circulant à 90 km/h et une voiture dont la vitesse varie de 110 à 130 km/h selon les expériences. Lorsque la voiture reste trop peu de temps à portée du RSU pour recevoir la totalité des fragments, le camion lui retransmet les fragments qui lui manquent après avoir dépassé la borne. Le détail de cet algorithme nommé n-ack (où n est un paramètre entier) est présenté à l'algorithme 5.

Algorithme 5 : n**-ack**

1 Initialisation :
  ▷ *Numéro de séquence du prochain fragment*
2 *nseq* ← 0

```
        ▷ Identifiants des fragments acquittés lors du prochain accusé de réception
3       ack-en-suspens ← ∅
        ▷ Messages non acquittés
4       messages-en-suspens ← ∅
5       armer le timer

6   Expiration du timer :
        ▷ Si suffisamment de messages ont été acquittés
7       si length(messages-en-suspens) < n alors
8           nseq ← nseq + 1
9           msg ← choix-fragment()                ▷ Prochain fragment à envoyer
10          messages-en-suspens[nseq] ← msg
11          envoyer( seq=nseq, payload=msg )
12      sinon
13          n ← select(messages-en-suspens)
14          envoyer( seq=n, payload=messages-en-suspens[n] )
15      fin si

16  Réception d'un fragment :
17      recevoir( seq=n, payload=m )
18      si n ∉ ack-en-suspens alors
19          append(ack-en-suspens, n)
20          si length(ak-en-suspens ≥ n alors
21              envoyer( messages_reçus = ack-en-suspens )
22              ack-en-suspens ← ∅
23          fin si
24      fin si

25  Réception d'un accusé de réception :
26      recevoir( messages_reçus=r )
27      pour chaque i ∈ r faire
28          delete(messages-en-usspens[i])
```

Dans le premier scénario, la voiture entreprend de dépasser le camion aux environs de la borne. Un autre scénario est étudié où la voiture circule dans le sens opposé au camion, qu'elle croise avant de rejoindre la borne. C'est dans ce cas du camion que la voiture reçoit ses premiers fragments de donnée alors que la borne complète par la suite.

Dans les deux cas, on choisit la propriété de performance algorithmique telle que la voiture doit recevoir la totalité de la donnée fixée à $N = 24$ fragments. Dans ce scénario nettement plus complexe, la propriété de performance topologique choisie est que la somme des capacités des n-arêtes entre la voiture et les autres nœuds (sans prendre en compte les n-arêtes reliant la voiture et le camion avant que ce dernier ne démarre les émissions de fragments) est supérieure au nombre de fragments composant la donnée. Les arêtes apparaissant dans les p-graphes dynamiques d'ordre inférieur à n ne sont pas prises en compte. En effet, elles n'assurent pas que le récepteur enverra un accusé de réception et donc, que l'émetteur sera mis au courant de la progression de la transmission.

On prend en compte à la place de la capacité des n-arêtes reliant la voiture et le camion la capacité des n-arêtes ayant relié la borne et le camion si cette dernière lui est inférieure (le camion n'a pas pu retransmettre des fragments qu'il n'a pas reçus). La propriété de performance topologique choisie est décrite à l'équation 6.6. Dans ce cas d'étude, la propriété de performance topologique n'est pas nécessaire à la réalisation de la propriété de performance algorithmique : avec de la chance, toutes les transmissions de fragments pourraient avoir lieu sur des p-arêtes ayant une capacité inférieure à n (mais peu d'accusés de réception auraient alors permis à la source de s'ajuster). Il ne s'agit pas non plus d'une propriété suffisante à la réalisation de la propriété de performance algorithmique : si le camion et la voiture n'ont pas manqué les mêmes fragments, il reste possible que la borne envoie un fragment redondant à la voiture afin de mettre à jour le camion, gâchant alors une partie des capacités des liens qui relient la borne et la voiture (il en est de même pour le camion).

$$PP_{n-ack} \equiv \min \left(\sum_{\substack{e\,:\,n-arete \\ (RSU,\,camion)}} C(e), \sum_{\substack{e\,:\,n-arete \\ (camion,\,voiture)}} C(e) \right) + \sum_{\substack{e\,:\,n-arete \\ (RSU,\,voiture)}} C(e) \geq N \tag{6.6}$$

La qualité des prédictions dans le scénario de dépassement est présentée sur la figure 6.8, sur laquelle apparaissent à la fois des succès applicatifs (environ 75 % des tests dans chaque scénario) et des échecs applicatifs. Parmi les échecs applicatifs, une part non négligeable n'a pas été prédite (20 à 50 % selon les scénarios). Ces erreurs de prédictions sont susceptibles d'affecter l'utilité de la prédiction dans un contexte de sécurité routière.

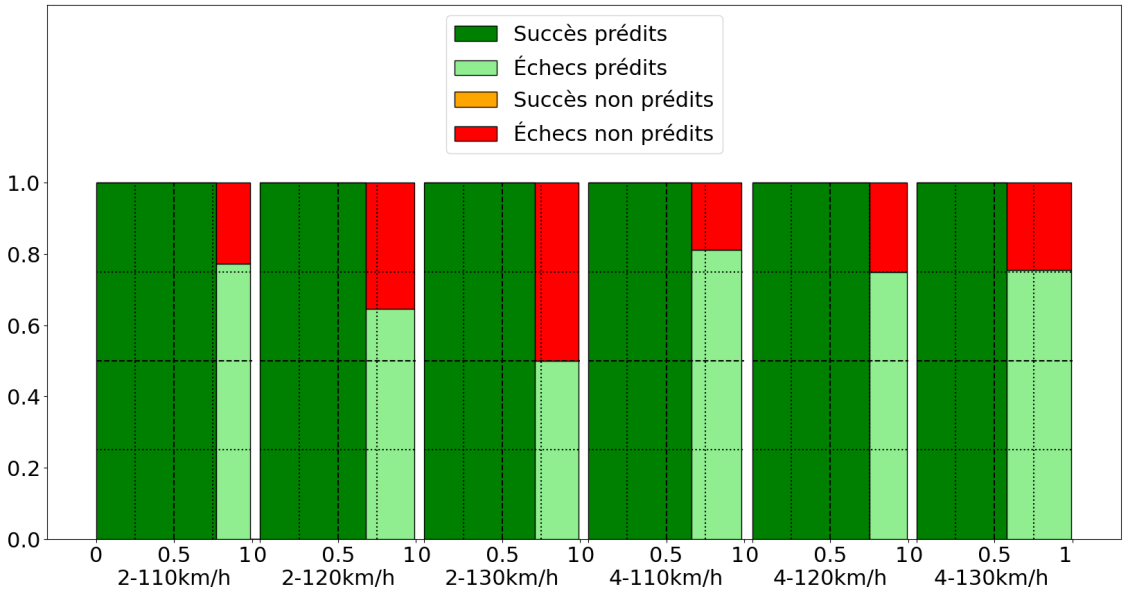


FIGURE 6.8 – Qualité des prédictions pour les expériences impliquant l'algorithme n-ack dans le scénario de dépassement autoroutier à différentes vitesses d'un camion roulant à 90 km/h. Plusieurs valeurs du paramètre n sont testées, elles apparaissent avant la vitesse dans la désignation du scénario.

La qualité des prédictions dans le scénario de croisement est présentée sur la figure 6.9, sur laquelle une large majorité des expériences (plus de 90 % dans tous les scénarios

sauf celui à 13km/h avec $n = 4$) se solde par un succès applicatif. Ces résultats rendent difficile l'interprétation des proportions d'échecs prédits et non prédits, qui est d'ailleurs très variable selon le scénario (entre 0 % et 100 %). Ils témoignent d'un choix de propriété de performance algorithmique trop simple pour ce scénario.

Parmi les succès et les échecs algorithmiques, des erreurs de prédictions se produisent, ce qui montre que la propriété de performance topologique n'est qu'une approximation de la propriété de performance algorithmique. Cela signifie qu'on ne peut pas assurer, même si on ne prédit que des succès algorithmiques, qu'aucun échec ne se produira (et inversement).

Ces résultats confirment bien la nécessité, pour pouvoir déduire des garanties de sécurité à partir de la méthode de prédiction proposée, de choisir comme propriété de performance topologique une propriété suffisante à la réalisation de la propriété de performance algorithmique. C'est cette approche qui sera développée dans la suite de ce travail.

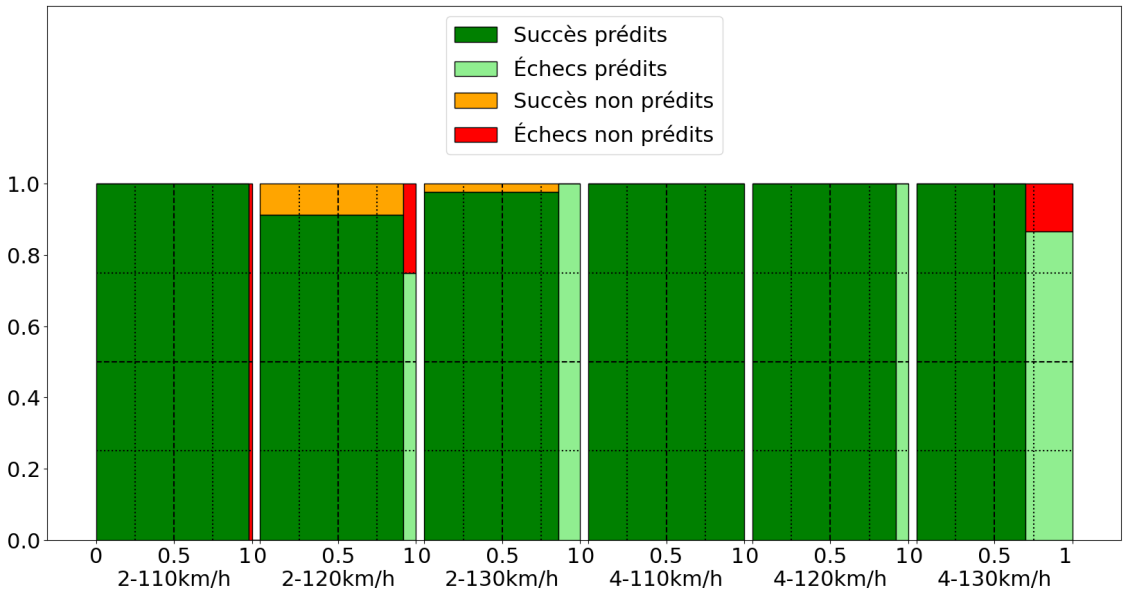


FIGURE 6.9 – Qualité des prédictions pour les expériences impliquant l'algorithme n-ack dans le scénario de croisement autoroutier à différentes vitesses d'un camion roulant à 90 km/h. Plusieurs valeurs du paramètre n sont testées, elles apparaissent avant la vitesse dans la désignation du scénario.

6.3.4 Diffusion

Dans un réseau véhiculaire, la diffusion d'information est l'une des opérations basiques les plus utilisées. Elle sert à la fois pour la diffusion d'alertes ou autres informations utiles, mais constitue aussi une composante importante de la plupart des algorithmes de coopération (consensus, élection, exclusion mutuelle, etc.). À partir de l'algorithme de diffusion PI proposé par [**?**], il est possible d'élaborer un algorithme de diffusion en réseau véhiculaire présenté à l'algorithme 6. Il s'agit principalement de faire suivre une seule fois le message à diffuser.

Algorithme 6 : **Propagation véhiculaire d'information sur le nœud** id

```
1   Initialisation :
2     message ← null
3     source ← id
4     propage ← False

5   Lancer la diffusion de la chaîne de caractère c :
      ▷ Création du
6     message ← c
7     source ← id

8   Réception d'un message d'un autre nœud :
9     recevoir( origin = s, payload = m )
10    si non propage alors
11      message ← m
12      source ← s
13    fin si

14  Expiration du timer :
15    Réarmer le timer
16    si message ≠ null et non propage alors
17      envoyer( origin = source, payload = message )
18      propage ← True
19    fin si
```

Cet algorithme est étudié dans un scénario urbain où un convoi de véhicules circule sur une même trajectoire dans la ville de Compiègne. À un instant aléatoire, le véhicule d'intérêt ($v1$) initie une diffusion destinée à tous les autres véhicules du convoi. La taille du convoi varie entre 6 et 11 véhicules qui démarrent aléatoirement sur leur trajectoire.

Dans ces circonstances, la propriété de performance algorithmique est la réception par chacun des véhicules du convoi, au moins une fois, de la donnée à diffuser. On peut alors en déduire la propriété de performance topologique selon laquelle il suffit qu'au moment où la diffusion est initiée, il suffit qu'il existe un entier p tel que le p-graphe est connexe et a un diamètre $d \leq p$. Comme les véhicules sont sur la même trajectoire et ne se dépassent pas, il est impossible qu'un véhicule dont tous les voisins sont encore en attente du message passe dans une situation où tous ses voisins ont déjà reçu le message et ne le rediffuseront plus. Il est donc possible de simplifier la propriété de performance topologique qui est alors que le 1-graphe soit connexe lorsque la diffusion est initiée.

La qualité des prédictions au cours de ces expériences est présentée sur la figure 6.10. On peut y remarquer que chaque scénario peut s'avérer être un succès ou un échec applicatif (67 % de succès à 6 véhicules, 50 % de succès à 9 et 30 % à 11). Si tous les échecs sont correctement prédits (absence d'échecs non prédits), quelques (moins de 5 % des tests) succès n'ont pas été correctement prédits. Ces résultats témoignent du fait que la propriété de performance topologique a été choisie suffisante mais pas nécessaire à la réalisation de la propriété de performance algorithmique (des déconnexions temporaires peuvent avoir lieu si les liens sont présents à l'exact bon moment).

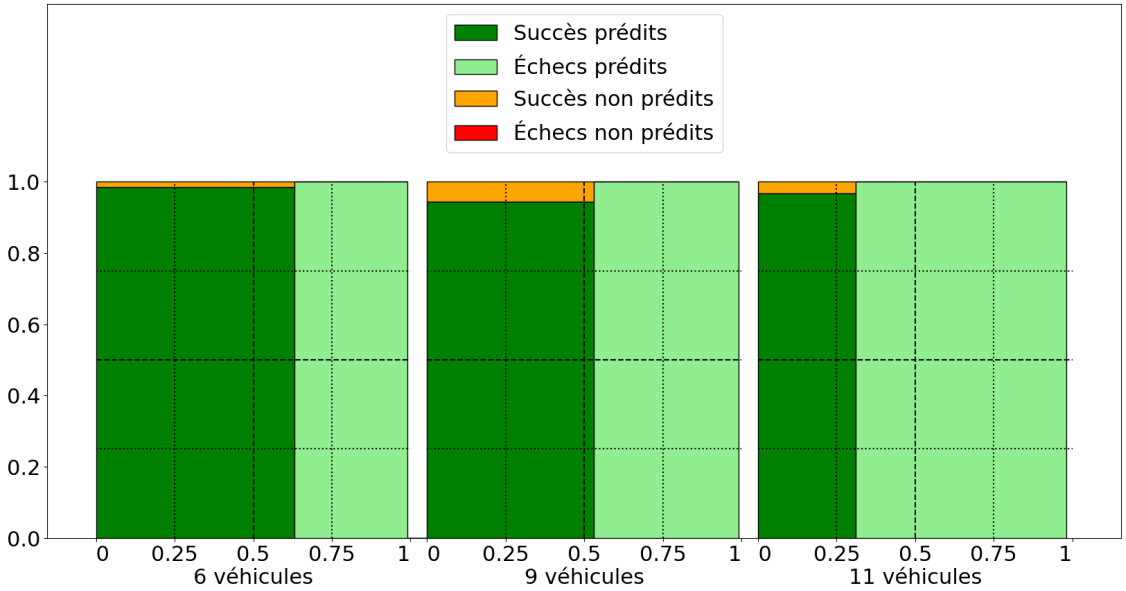


FIGURE 6.10 – Algorithme de diffusion PI dans un scénario urbain impliquant 6,9 et 11 véhicules

Les résultats expérimentaux valident la méthode de prédiction dans un contexte applicatif complexe, avec un algorithme réaliste à la base de nombreux algorithmes de coopération. Ils ont été obtenus au cours d'un scénario réaliste quoi que peu complexe (changements topologiques limités) et méritent d'être confirmés dans d'autres cas d'usages.

6.3.5 Diffusion avec accusé de réception

Une version avec accusé de réception de l'algorithme PI est proposé dans [**?**], puis adaptée dans [**?**] sous le nom dPIF aux réseaux dynamiques. Elle peut notamment être utilisée pour effectuer une collecte d'informations locales ou pour découvrir le réseau (et sa topologie) à plusieurs sauts.

Cet algorithme est étudié dans un scénario urbain où deux convois de véhicules sont sur des trajectoires qui se croisent. Au cours de l'expérience, le véhicule d'intérêt (v_1) initie une diffusion et attend les retours des autres nœuds. La taille des convois varie de 2 véhicules (donc 4 véhicules dans l'expérience) à 4 véhicules (donc 8 véhicules dans l'expérience). L'algorithme est paramétré pour diffuser jusqu'à $d = 3$ sauts. Dans ces circonstances, l'algorithme est considéré couronné de succès si le véhicule d'intérêt reçoit un accusé de réception de la part de chacun des autres véhicules de l'expérience. Cette proposition constitue la propriété de performance algorithmique et l'on peut en déduire la propriété de performance topologique selon laquelle il suffit que le $2*d+2$ graphe soit connexe et dispose d'un arbre couvrant enraciné en v_1 dont la profondeur est inférieure à 3 pour assurer la propriété de performance algorithmique. En effet, cette propriété assure la stabilité d'un arbre couvrant sur une durée suffisante pour assurer à la fois la diffusion et le retour.

La qualité des prédictions au cours de ces expériences est présentée sur la figure 6.11. On peut y remarquer que chaque scénario peut s'avérer être un succès ou un échec

applicatif (30 % de succès à 4 et 6 véhicules, et 60 % à 8). L'absence complète d'échecs non prédits témoigne d'un choix de propriété de performance topologique suffisante pour assurer la propriété de performance algorithmique. Cependant, la proportion de succès non prédits est plutôt importante (environ la moitié des succès pour les scénarios à 4 et 6 véhicules, 75 % des succès pour le scénario à 8 véhicules). Cela pourrait être interprété comme une prudence excessive de la propriété de performance topologique. En effet, il est par exemple possible que le lien soit rompu entre l'initiateur et l'un de ses voisins entre la réception du message par ce voisin et l'envoi de son accusé de réception sans impacter la propriété de performance algorithmique. Il serait donc possible d'affiner la propriété de performance topologique en en tenant compte si, dans une prédiction réelle, les succès prédits n'étaient pas suffisamment majoritaires.

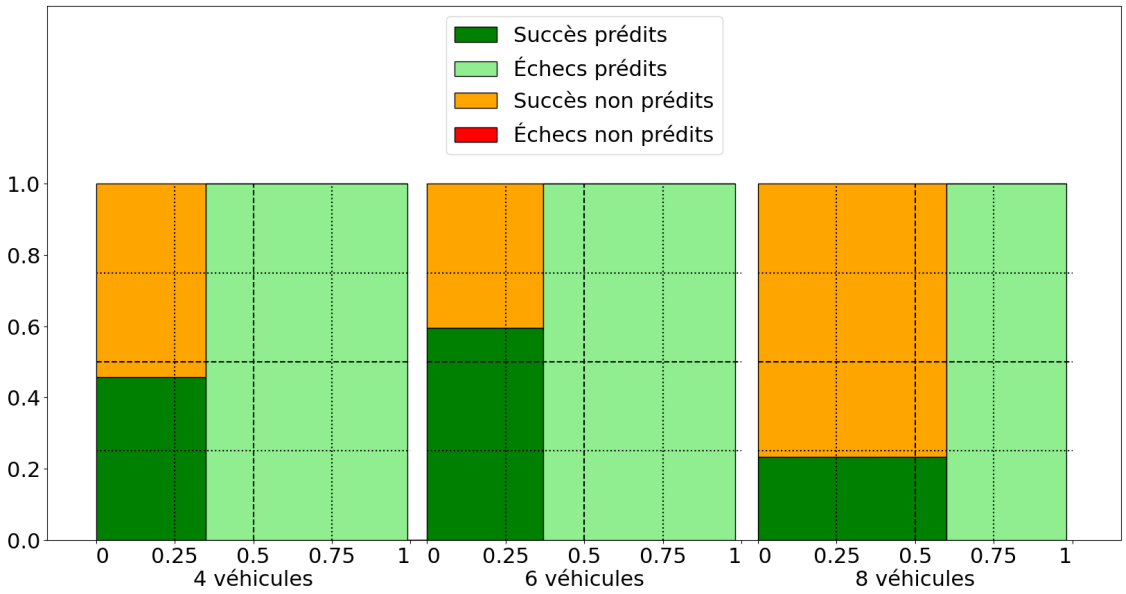


FIGURE 6.11 – Qualité des prédictions dans les expériences impliquant l'algorithme de diffusion avec accusé de réception PIF en scénarios urbains impliquant 4,6 et 8 véhicules.

6.3.6 Découverte de voisinage

En réseau véhiculaire, ad hoc, la coopération est généralement locale, c'est-à-dire qu'elle implique des véhicules à portée de communication ou à quelques sauts au maximum. Connaître les véhicules avec lesquels on peut communiquer est d'une importance capitale, raison pour laquelle les standard concernant les messages communiqués comprennent des messages d'identification périodique (CAM ou BSM [**?**, **?**], notamment).

Un algorithme de découverte de voisinage basé sur l'algorithme permettant le choix des Multi-Point Relays (MPR) dans le protocole OLSR [**?**] est décrit à l'algorithme 7 sous le nom PND (*Proactive Neighborhood Discovery*). Il permet à chaque nœud de maintenir à jour une liste de ses voisins. Cette liste peut être utilisée par d'autres applications, par exemple pour envoyer des messages unicast, faire du routage à quelques sauts ou encore établir des groupes de véhicules proches. L'exécution de cette algorithme consiste, pour chaque nœud du réseau, à envoyer régulièrement une balise indiquant son identifiant et sa liste de voisins détectés. À réception d'une balise, le nœud récepteur ajoute la source

de l'émission à ses voisins détectés et cherche si une communication bidirectionnelle a été établie.

Algorithme 7 : **Proactive neighborhood discovery sur le nœud *id***

1 Initialisation :
  ▷ *Nœuds détectés à travers une communication bidirectionnelle*
2 *voisins* $\leftarrow \emptyset$
  ▷ *Nœuds détectés sans preuves de communication bidirectionnelle*
3 *voisins_entendus* $\leftarrow \emptyset$

4 Expiration du timer :
5 Réarmer le timer
6 **envoyer**(src = *id*, 1D = *voisins_entendus* , 2D = *voisins*)

7 Réception d'un message de PND sur un autre nœud :
8 **recevoir**(src = *sndr*, 1D = $n1$, 2D = $n2$)
  ▷ *Si l'expéditeur a détecté le nœud courant*
9 **si** $id \in n1$ **ou** $id \in n2$ **alors**
10 *voisins* $\leftarrow$ *voisins* $\cup \{sndr\}$
11 **fin si**
12 *voisins_entendus* $\leftarrow$ *voisins_entendus* $\cup \{sndr\}$

Cet algorithme est étudié dans des scénarios urbains où deux convois de plusieurs véhicules circulent sur deux trajectoires différentes, certains à 30 km/h, d'autres à 40 km/h. Une illustration de ce scénario est disponible sur la figure 6.12 Tous les véhicules exécutent l'algorithme et enregistrent donc leurs voisins en permanence et démarrent tous aléatoirement sur leur trajectoire. L'expérience dure quelques secondes pour permettre quelques échanges de messages (la période d'émission est fixée à 1 s). L'essai est ensuite arrêté.

Dans ces circonstances, la propriété de performance algorithmique choisie est qu'à la fin de l'expérience, chacun des véhicules a détecté au moins 3 voisins. La propriété de performance topologique est alors que chacun des nœuds du dernier 2-graphe appartenant au 2-graphe dynamique a un degré d'au moins 3. En effet, si deux nœuds sont voisins dans le 2-graphe dynamique, ils se détectent forcément l'un l'autre : s'ils sont voisins dans le 2-graphe quand l'un des deux émet, son message arrive forcément à l'autre. La 2-arête qui les relie permet l'envoi d'un second message. En raison des émissions périodiques, il s'agira du second nœud qui enverra un message reçu par le premier.

La qualité des prédiction au cours de ces expériences est présentée sur la figure 6.13. On peut y remarquer que la difficulté de la tâche diminue avec le nombre de véhicules impliqués (un réseau plus dense a un degré moyen plus élevé sur chaque nœud). La proportion de succès passe ainsi de 5 % à 6 véhicules à 25 % à 8, 50 % à 10 et 55 % à 12 véhicules. L'absence complète d'échecs non prédits, valide les capacités de la méthode de prédiction proposée à apporter des garanties en termes de sécurité routière lorsque la propriété de performance topologique est suffisante à la propriété de performance algorithmique. Si la proportion de succès non prédits reste importante, on remarque qu'elle diminue lorsque la proportion de succès augmente, ce qui signifie que dans la situation recherchée par le prédicteur (lorsque la propriété de performance topologique

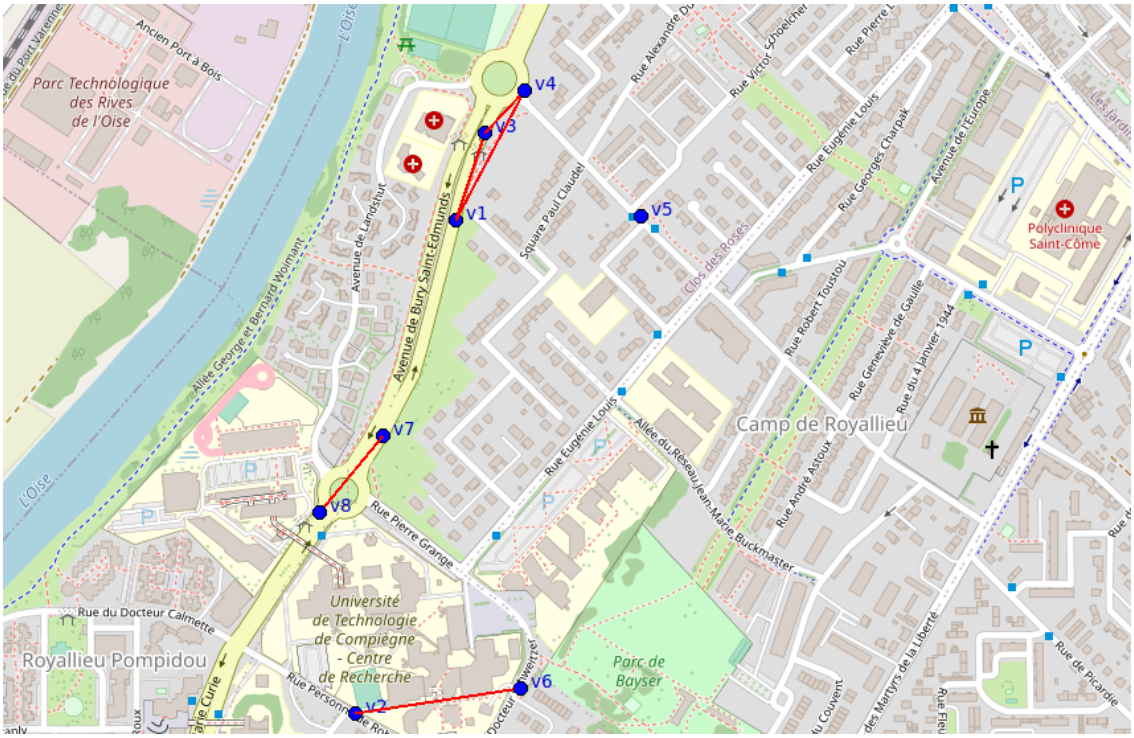


FIGURE 6.12 – Capture d'écran de l'émulateur au cours de l'exécution d'une expérience concernant l'algorithme PND utilisant le scénario urbain à 8 véhicules sur 2 trajectoires. Les liens en rouge représentent les communications disponibles

est quasiment toujours satisfaite, donc lorsque l'immense majorité des prédictions sont des succès), la prédiction est particulièrement fiable.

6.3.7 Détection des triangles

Lorsqu'un message doit être diffusé à plusieurs sauts, il peut parfois être utile de restreindre les nœuds qui vont le relayer. En effet, s'il existe un nœud v, voisin de l'initiateur u de la diffusion, tel que tous les voisins de v sont également des voisins de u, la retransmission par le nœud v n'apporte rien par rapport à la transmission par le nœud u. Il peut donc s'avérer utile de détecter ces situations-là, qui se traduisent par l'existence d'un triangle dans la topologie réseau.

Un algorithme de détection des triangles basé sur l'algorithme 7 de découverte de voisinage PND est présenté à l'algorithme 8 et nommé PTD (Proactive Triangles Detection). Il permet à chaque nœud du réseau de maintenir à jour la liste des triangles auxquels il participe grâce aux messages reçus de ses voisins.

Algorithme 8 : **PTD sur le nœud** *id*

1 Initialisation :
  ▷ *Nœuds détectés sans preuves de communication bidirectionnelle*
2 $voisins \leftarrow \emptyset$
  ▷ *Triangles détectés*
3 $triangles \leftarrow \emptyset$

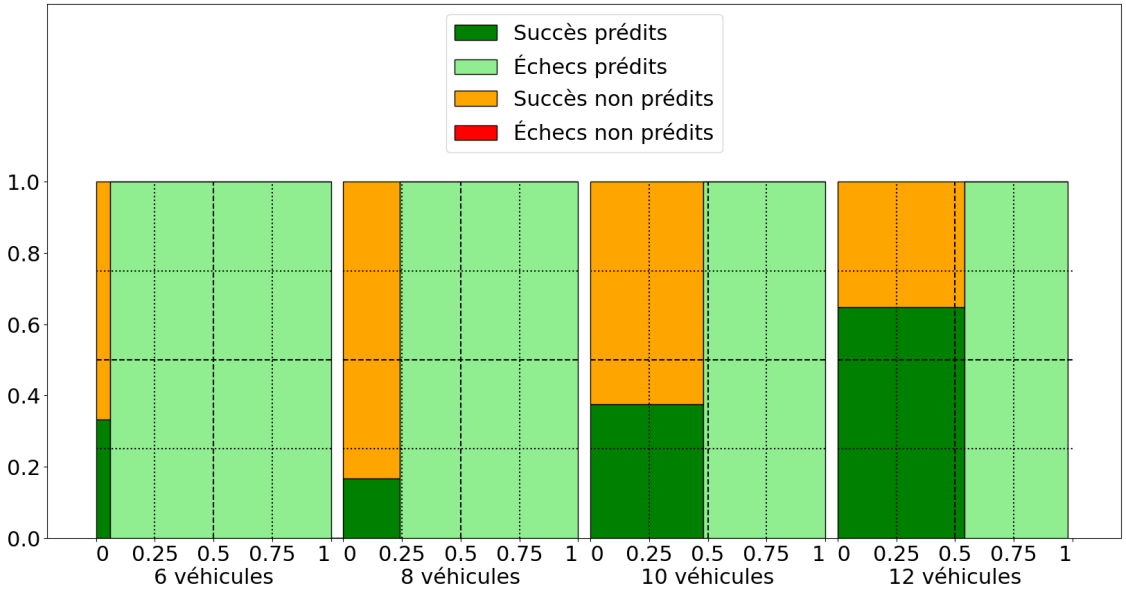


FIGURE 6.13 – Qualité des prédictions dans les expériences impliquant l'algorithme de découverte de voisinage PND dans des scénarios urbains impliquant 6, 8, 10 et 12 véhicules

4 Expiration du timer :
5 Réarmer le timer
6 **envoyer**(src = *id*, voisinage = *voisins*)

7 Réception d'un message de PTD sur un autre nœud :
8 **recevoir**(src = *sndr*, voisinage = v)
9 $voisins \leftarrow voisins \cup \{sndr\}$
10 **pour chaque** $u \in v$ **faire**
11 **si** $u \in voisins$ **alors**
12 $triangles \leftarrow triangles \cup \{\{id, u, sndr\}\}$
13 **fin si**

Cet algorithme est étudié dans des scénarios urbains identiques à ceux utilisés pour l'étude de l'algorithme PND, décrits à la section 6.3.6, où deux convois de véhicules ayant des vitesses différentes circulent sur deux trajectoires urbaines qui se croisent.

Dans ces circonstances, la propriété de performance algorithmique sera différente en fonction du nombre de véhicules impliqués dans l'expérience. Pour le scénario à 6 véhicules, la propriété de performance algorithmique est l'appartenance de chaque véhicule à au moins 4 triangles tandis que pour le scénario à 8 véhicules, il s'agit de l'appartenance de chaque nœud à au moins 11 triangles. Lorsque le scénario implique 10 véhicules, ils doivent tous appartenir à au moins 22 triangles pour valider la propriété de performance algorithmique et pour le scénario impliquant 12 véhicules, le nombre de triangles auquel chaque nœud doit appartenir pour la remplir est 50.

La qualité des prédictions au cours de ces expériences est représentée sur la figure 6.14. On peut y remarquer à nouveau l'absence totale d'échecs non prédits.La proportion de succès correctement prédits est très importante (plus de 80 %). La proportion de succès réels est similaire pour tous les scénarios (environ 50 %) ce qui s'explique par un

Nombre de véhicules	Triangles détectés (propriété de performance algorithmique)	Triangles dans le dernier 2-graphe de l'expérience
6	4	4
8	11	11
10	22	22
12	50	50

TABLE 6.1 – Synthèses des propriétés de performance algorithmiques et topologiques par scénario étudié

choix de propriété de performance algorithmique adapté à chaque scénario. Lors d'une prédiction réelle destinée à décider du déploiement ou de l'abandon d'une application véhiculaire, la propriété de performance algorithmique est au contraire choisie en fonction de l'usage attendu de l'application.

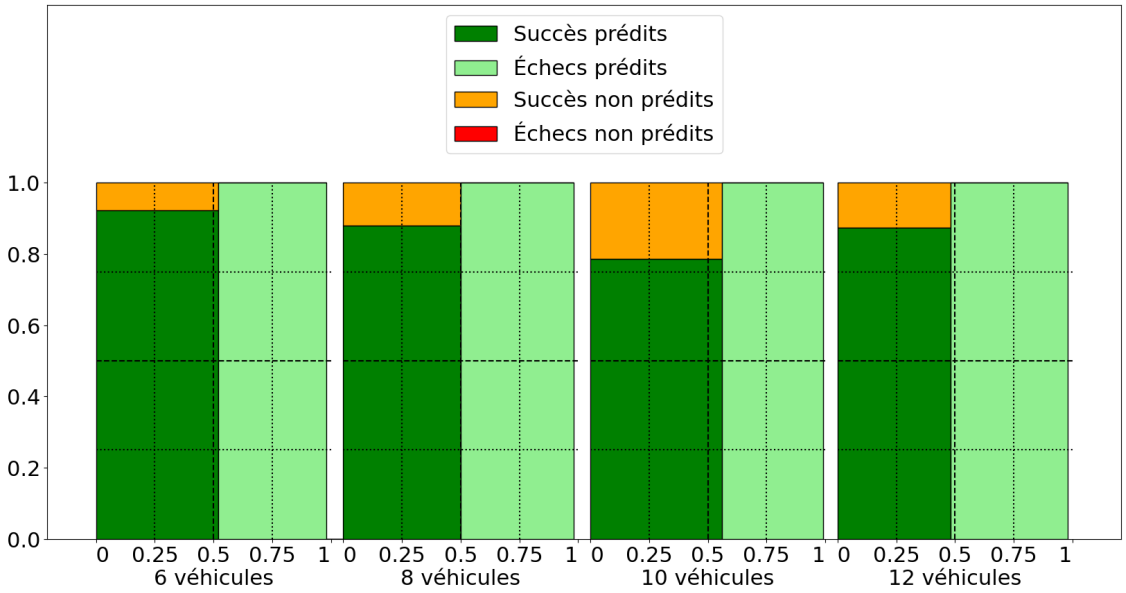


FIGURE 6.14 – Qualité des prédictions dans les expériences impliquant l'algorithme de détection des triangles PTD dans des scénarios urbains impliquant 6, 8, 10 et 12 véhicules

Ces résultats valident plus avant la méthode de prédiction proposée, avec des scénarios réalistes et un algorithme répondant à des problématiques réelles, inspiré de protocoles couramment utilisés pour les réseaux ad hoc.

6.4 Discussion

Les expériences menées pour tester la méthode de prédiction proposée montrent qu'elle a de bonnes capacités prédictives dès lors que la propriété de performance topologique est suffisamment proche de la propriété de performance algorithmique ou que leur relation est caractérisée (propriété nécessaire ou propriété suffisante). Ces capacités prédictives peuvent fournir suffisamment de garanties en termes de sécurité routière

pour envisager le déploiement d'une application, résultat aujourd'hui très difficile à obtenir par d'autres moyens.

Obtenir la propriété de performance topologique est cependant un travail relativement difficile (moins qu'une preuve, cependant) d'analyse d'algorithme, qui doit être renouvelé sur chaque algorithme étudié (et même possiblement à chaque mise à jour affectant significativement le fonctionnement de l'algorithme). De plus, rien ne garantit qu'il est possible de trouver une propriété suffisante à la propriété de performance algorithmique et suffisamment proche de la réalité pour que les prédictions aboutissent à une très large majorité de succès dans des conditions réelles, et ce même si des tests empiriques confirment que l'application fonctionne dans ces conditions.

Hormis dans le cas de l'algorithme simple Beacon pour des scénarios autoroutiers, la propriété de performance topologique s'est toujours révélée significativement éloignée de la propriété de performance algorithmique, notamment à cause du modèle utilisé qui discrétise le temps et fait perdre de l'information sur les évolutions topologiques. Cela peut rendre l'utilisation d'autres modèles nécessaires dès lors qu'une propriété nécessaire et suffisante est souhaitée.

La propriété de performance topologique est généralement exprimée en utilisant le modèle des p-graphes dynamiques, qui met l'accent sur les coopérations restreintes à la fois localement et temporellement. D'autres modèles peuvent s'avérer nécessaire lorsque ces conditions ne sont pas remplies (réseau tolérant les délais *DTN*, par exemple).

La méthode proposée fait l'hypothèse qu'il est possible de capter et modéliser un trafic suffisamment représentatif des conditions réelles pour que les garanties offertes correspondent pleinement à la réalité ce qui pourrait ne pas être systématiquement le cas (lorsque le trafic varie énormément au cours de la journée, de l'année, etc.).

Des expériences de validations de la méthode sur route sont envisagées pour la suite, afin d'apporter une mise à l'épreuve en conditions réelles de la méthode, mais aussi d'étudier la faisabilité technique de la capture de p-graphes dynamiques sur trafic réel. Elles seront également l'occasion de comparer les résultats émulés aux résultats réels, donc de réaliser des expériences sur de nouveaux scénarios (proches des scénarios réels) en émulation.

6.5 Conclusion

La méthode de prédiction proposée est capable d'offrir des garanties attendues par les exploitants de réseaux routiers sur l'effet sur la sécurité routière du déploiement applicatif. En effet, en choisissant uniquement des propriétés de performances topologiques suffisantes pour assurer la propriété de performance algorithmique correspondante, on peut obtenir des garanties de bon fonctionnement de l'algorithme sur un trafic donné. Si ce trafic est représentatif, ces garanties ont des applications réelles en termes de sécurité routière.

Cependant, la méthode proposée ne semble pas pouvoir être entièrement automatisée et le modèle utilisé pour représenter le scénario peut avoir un impact sur les scénarios et algorithmes qui peuvent être étudiés.

Par ailleurs, la méthode n'a pas vocation à se substituer aux preuves algorithmiques, qui, si elles sont plus difficiles à obtenir, peuvent s'avérer plus complètes et cherchent à assurer plusieurs propriétés (sûreté, vivacité) en même temps, dans un modèle théorique (habituellement moins réaliste) de système réparti. Ces dernières sont notamment

importantes en termes de sécurité informatiques, puisqu'elles identifient les possibles failles applicatives.

Chapitre 7

Application à un problème concret de sécurité routière

Sommaire

7.1 Introduction

Alors que la méthode de prédiction proposée au chapitre 5 a montré un certain réalisme dans des situations théoriques au chapitre 6, une étude dans un cas d'usage plus concret semble nécessaire. En effet, cette dernière permettrait de montrer l'utilité d'une telle prédiction pour améliorer la sécurité routière et de valider la faisabilité de toute la chaîne de traitement à effectuer.

L'une des approches les plus prometteuses en termes d'amélioration de la sécurité routière est de développer des systèmes de protection destinés aux usagers les plus vulnérables sur les routes. En effet, ils représentent près de 20 % des décès suite à un accident routier qui sont causés à 70 % par des véhicules légers (voitures) [**?**]. Ce cas d'accidents (collision piéton voiture) est de plus en plus étudié par la communauté scientifique, et

représente, du fait de la mortalité qu'il engendre, l'un des principaux enjeux de déploiement des C-ITS pour la sécurité routière. C'est donc sur ce cas d'étude que se concentrera l'analyse de la prédiction comme outil de validation dans un cas réel de déploiement.

Afin de réaliser cette étude, une description des enjeux de la protection des piétons est tout d'abord proposée avec un état de l'art des systèmes de sauvegarde des plus vulnérables. Cela permet ensuite de déterminer le scénario précis sur lequel la prédiction se concentre avant de conduire l'étude expérimentale.

7.2 État de l'art

Les usagers vulnérables des routes sont des personnes plus exposées aux dangers de la circulation en raison de leur faible protection physique et mécanique. Ils représentent donc un enjeu majeur de sécurité routière pour lequel la communauté scientifique montre un intérêt particulier.

7.2.1 Classification des usagers vulnérables (VRU)

Les usagers vulnérables des routes constituent une catégorie diverse d'utilisateurs du réseau routier. L'ETSI propose une norme [**?**] définissant les VRU (*Vulnerable Road users* ou Usagers Vulnérables des Routes) et les classe sous 3 profils. Dans ce cadre, nous détaillons les différents usagers des routes considérés comme vulnérables.

Les piétons sont les usagers des routes les plus exposés aux risques, car ils ne disposent d'aucune protection sur eux, ne peuvent circuler à grande vitesse et ont parfois une faible manœuvrabilité (personnes âgées, ou transportant des objets encombrants, par exemple).

Les piétons bénéficient de voies de circulation dédiées (les trottoirs), permettant de voyager à travers la plupart des agglomérations sans grand danger. Leur faible allure leur permet habituellement de s'arrêter presque instantanément, ce qui peut participer à leur sécurité (distance parcourue pendant le temps de réaction nulle). Les règles de circulation donnent priorité aux piétons sur tous les autres usagers de la route.

Ainsi, l'architecture de l'infrastructure routière comme les règles de circulation sont conçues pour protéger ces utilisateurs les plus vulnérables des routes, même lorsqu'ils ne tiennent aucun compte des règles de circulation et ne font aucun effort pour se protéger eux-mêmes (signaux lumineux, casque).

Les utilisateurs de mobilités douces non motorisées (trottinette, rollers, skateboard) ou faiblement motorisées (trottinette électrique, hoverboard) sont également vulnérables en raison de leur manque de protection physique.

Ces usagers utilisent à la fois les voies de circulation dédiées aux piétons (trottoirs) et aux véhicules, ce qui les met encore plus en danger. Leur protection légale est, en France identique à celle des piétons, malgré une allure supérieure et une manœuvrabilité parfois plus faible.

Les cyclistes sont des utilisateurs vulnérables, disposant d'une capacité d'arrêt et d'une manœuvrabilité plus limitée que celles de piétons, malgré une vulnérabilité similaire.

Leur vélo ne dispose habituellement d'aucun système de protection malgré quelques systèmes de prévention (catadioptres, phares) mais le cycliste peut être équipé de systèmes de protection additionnels (casques, gants) non obligatoires. La circulation des

cyclistes est soumise aux règles des véhicules et s'effectue sur les mêmes voies ou des voies dédiées (pistes cyclables).

Enfin, les véhicules faiblement motorisés tels que les cyclomoteurs sont considérés comme des usagers vulnérables en raison de la faiblesse des protections mécaniques et de leur allure limitée. Ils possèdent cependant beaucoup de caractéristiques communes avec les véhicules (voies et règles de circulation, présence d'une source d'énergie et d'équipements de protection obligatoires).

Tous les VRU ne peuvent pas être équipés de systèmes de communication véhiculaires. En effet, comme l'identifie la norme ETSI [**?**], ils ne disposent pas tous d'une source d'énergie fiable (notamment les piétons et les mobilités douces) tandis que leur structure est souvent trop accessible de l'extérieur et fragile (vélos, skateboards) pour protéger des équipements de communication. Les véhicules faiblement motorisés pourraient en revanche être équipés d'un dispositif de communication identique à celui des véhicules. Ils s'intégreraient ainsi au réseau véhiculaire de la même manière que tous les autres véhicules. Du fait des différences fondamentales entre les VRU, les études de cas se concentrent généralement sur certains d'entre eux, qui pourraient se définir par les profils identifiés dans la norme (profil 1 pour les piétons, profil 2 pour les cyclistes et mobilités douces, et profil 3 pour les véhicules faiblement motorisés), même si l'intégration des 3 profils au C-ITS est finalement visée.

7.2.2 Classification des architectures de coopération

Les VRU circulent à la fois en ville et sur les routes de campagne. Ils sont donc sujet à de nombreux risques différents, qui ne peuvent pas tous être gérés de la même manière par une coopération entre usagers de la route. Différents cas d'usage et stratégies sont étudiés dans la littérature afin de protéger les utilisateurs vulnérables. Ils ont été identifiés et classifiés par ailleurs par la norme ETSI TR-103-300-1 [**?**].

Coopération entre plusieurs VRU

Puisque les piétons circulent habituellement sur une voie qui leur est réservée, et n'ont pas forcément un entraînement à la vigilance particulier (examen d'aptitude, par exemple), le risque de collision entre plusieurs VRU peut être à prendre en compte.

Ce cas d'usage nommé A propose d'utiliser les téléphones des VRU afin de détecter les risques de collision entre VRU. S'il semble irréaliste et peu utile de le mettre en œuvre pour éviter les collisions entre piétons, il reste néanmoins envisageable de mettre en place une telle coopération pour les collisions impliquant les véhicules faiblement motorisés (profil 3) ou des cyclistes (profil 2). En effet, ce type de VRU est à la fois plus dangereux (vitesse et masse plus importantes) et plus faciles à équiper (source d'énergie).

Cependant, si les cyclomoteurs sont considérés de la même manière que les autres véhicules au sein du réseau véhiculaire, ce cas d'usage perd la quasi-totalité de son intérêt au profit des cas d'usage impliquant des véhicules. En effet, les usagers du profil 2 sont plus difficiles à équiper d'aides à la conduite et de systèmes de communication.

C'est la raison pour laquelle la littérature ne s'intéresse pas à ce type d'architectures, même si elle étudie la possibilité de faire coopérer les piétons pour la transmission d'information comme dans [**?**]. Avec cette approche, les piétons coopèrent pour déterminer

un leader qui est chargé des échanges avec le réseau véhiculaire. Cependant, l'application proposée ne permet de prévenir que des collisions impliquant un véhicule.

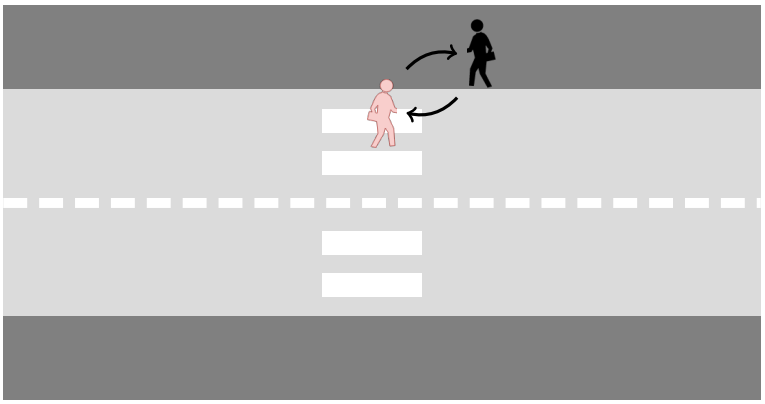


FIGURE 7.1 – Schéma de l'architecture de coopération entre VRU normée par l'ETSI (cas d'usage A).

Coopération directe entre VRU et véhicule

Éviter une collision impliquant un véhicule et un VRU peut intuitivement être réalisé en faisant coopérer les deux protagonistes le plus directement possible. Ce cas d'usage nommé B a notamment l'avantage d'éviter la latence éventuellement liée aux nœuds intermédiaires, et de ne pas nécessiter de perception de l'environnement (chacun peut se contenter de sa localisation si elle est suffisamment précise).

Une telle architecture de coopération suppose cependant que les VRU sont équipés d'un smartphone, qu'il est allumé et que les fonctionnalités nécessaires (communication, localisation) sont activées. Ces hypothèses ne sont pas toujours remplies sur une route ouverte (cas des enfants sans téléphone, des téléphones déchargés ou éteints).

Par ailleurs, la communication directe entre le VRU et le véhicule suppose un système de communication compatible. Ce défi technologique n'est pas si simple à résoudre, par exemple si les véhicules utilisent le WiFi véhiculaire 802.11p pour communiquer.

L'approche de coopération directe est celle retenue [**?**], où l'architecture de coopération est basée sur des smartphones capables d'utiliser le 802.11p, et l'émission de messages périodiques par les piétons et les véhicules. Lors de la réception d'un message périodique, chaque acteur calcule s'il risque une collision avec son expéditeur. Si un risque de collision est détecté, il lui retourne une notification, et prend localement les mesures appropriées (avertissement du conducteur ou VRU, freinage d'urgence). L'attitude attendue du VRU lorsqu'il reçoit un tel avertissement n'est pas précisée, tout comme l'interface homme machine qui permet cette transmission d'alerte. La proposition va ainsi plus loin que la norme en proposant d'avertir le VRU du risque de collision au même titre que le véhicule.

Une application pour smartphone complète basée sur l'architecture B est développée par [**?**] pour effectuer des tests sur route. Le risque de collision est estimé grâce à la distance longitudinale (distance à parcourir sur la route) qui sépare le piéton et le véhicule. Une alerte est émise à la fois sur le véhicule et le piéton via une interface homme machine décrite si le risque de collision est élevé. La technologie de communication utilisée

est également le 802.11p, même s'il peut provoquer une forte consommation d'énergie et n'est habituellement pas compatible à l'heure actuelle avec la plupart des smartphones. Une application similaire est décrite dans [**?**], mais avec le VRU supposé passif, c'est-à-dire qu'il ne reçoit pas les alertes et n'est pas supposé y réagir. Leur travail propose notamment des fonctionnalités pour limiter la consommation d'énergie liée à la localisation du téléphone. Ces fonctionnalités sont notamment basées sur une adaptation de la fréquence de mise à jour en fonction du mouvement du VRU.

L'application SaferCross, proposée dans [**?**] propose d'utiliser plutôt une variante du WiFi classique appelée WiFi Direct, qui accélère l'association afin de permettre une communication directe entre les véhicules et les piétons pour générer des alertes chez tous les acteurs. Cette application est tributaire de la portée limitée du WiFi Direct, semblable au WiFi domestique (un peu plus de 100 m) ce qui limite son usage. C'est une architecture qui avait été précedemment proposée par [**?**], à travers l'application V2ProVu, utilisant la technologie WiFi pour les communications entre véhicules et piétons, en considérant les VRU passifs. Les expériences proposées utilisent une tablette tenue dans les mains des piétons, et incluent une étude de l'impact du corps d'un piéton sur la portée du signal.

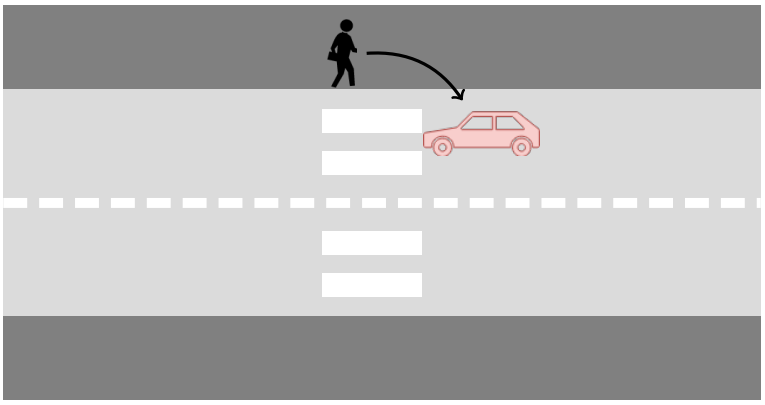


FIGURE 7.2 – Schéma de l'architecture de coopération directe entre VRU et véhicule, normée par l'ETSI (cas d'usage B).

Coopération entre véhicules

La plupart des véhicules neufs disposent de perceptions de leur environnement, même rudimentaires (caméras, détection de lignes blanches, radar de recul, etc.) qui pourraient être utilisés pour repérer les piétons en situation dangereuse. La coopération entre véhicules permet d'étendre la portée de ces perceptions, et d'assurer une meilleure sécurité aux piétons, indépendamment de leur équipement à travers le cas d'usage nommé C de la norme ETSI.

Cette approche prend comme hypothèse de départ qu'un véhicule est témoin de la scène dangereuse et capable d'avertir le véhicule à risque de collision avec le VRU. Cette hypothèse suppose donc, si la présence du véhicule témoin est nécessaire pour éviter une collision, que les véhicules en circulation ont une densité suffisante, mais également une forte pénétration des technologies permettant la détection embarquée de piétons, en plus des communications. Par ailleurs, le véhicule à risque de collision doit faire confiance au véhicule témoin (qui peuvent être plusieurs) suffisamment pour prendre des contre-mesures efficaces.

La littérature n'est pas très abondante sur cette architecture de coopération. C'est probablement lié aux hypothèses qu'elle nécessite sur la densité de trafic et la pénétration des systèmes de détection de piétons au sein des véhicules. Elle est cependant envisagée dans [**?**], complémentée par de nombreux autres mécanismes (communications P2V, perceptions locales du véhicule), destinés à de la fusion de donnée pour des véhicules autonomes. Dans [**?**], cette architecture est également prévue, mais en complément d'une détection par l'infrastructure routière. L'architecture de coopération C proposée par la norme ETSI [**?**] est plutôt étudiée en complément d'un système global incluant d'autres approche de protection des VRU.

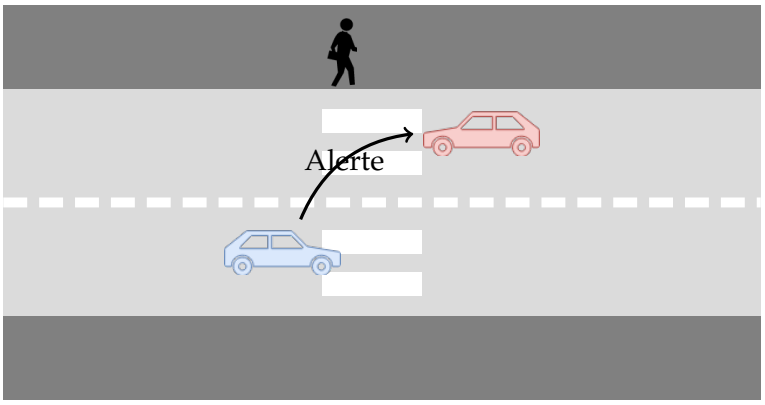


FIGURE 7.3 – Schéma de l'architecture de coopération entre véhicules normée par l'ETSI (cas d'usage C).

Coopération entre infrastructure et véhicule

Lorsqu'on connaît les zones où les piétons courent un risque particulier, il est possible d'équiper l'infrastructure routière de capteurs et calculateurs à même de détecter les situations dangereuses. En utilisant un système de communication compatible avec celui des véhicules de la zone (DSRC, par exemple), il est possible de les prévenir des dangers afin qu'ils prennent les contre-mesures adaptées.

Une solution basée sur une détection du danger par l'infrastructure n'a pas besoin de coopérer avec les piétons et est donc particulièrement adaptée si les piétons ne sont pas supposés disposer de téléphone intelligent ou les éteignent. Cela peut par exemple servir à sécuriser les abords d'une école.

Le déploiement d'une telle architecture (nommée D par la norme ETSI) et sa maintenance sont très coûteux. Pour être efficace, il doit donc avoir lieu dans une zone particulièrement dangereuse où de nombreux accidents impliquant des piétons ont habituellement lieu et pourraient donc être évités. Ainsi, c'est une architecture qui a sûrement plus sa place en milieu urbain, où le trafic véhiculaire et piéton est suffisant pour générer de tels risques. La gestion de la confiance est plus simple dans cette architecture, puisqu'il suffit que le gestionnaire de réseau routier fournisse une authentification aux alertes qu'il détecte pour en assurer la fiabilité (la détection est centralisée par l'infrastructure routière).

L'architecture D est par exemple choisie dans [**?**], où une caméra est utilisée pour détecter les piétons qui traversent à une intersection malgré le passage des véhicules. Les

piétons sont identifiés sur les images de la caméra par des techniques d'apprentissage profond et l'alerte est donnée à travers un message DENM [**?**] lorsque la situation est dangereuse.

D'autres travaux utilisent cette architecture, comme [**?**, **?**, **?**], le dernier complémente par ailleurs l'approche où la détection est réalisée par l'infrastructure avec une détection par les véhicules en circulation, selon le cas d'usage C de la norme ETSI.

La détection grâce à la perception de l'infrastructure est une approche de protection des piétons qui semble efficace, et qu'il est simple de compléter avec des approches comme la communication entre les piétons et l'infrastructure ou la fusion avec les données des capteurs embarqués dans les véhicules [**?**]. Elle semble adaptée pour former la base d'une solution collaborative de protection des piétons, en s'attaquant en priorité aux zones les plus dangereuses.

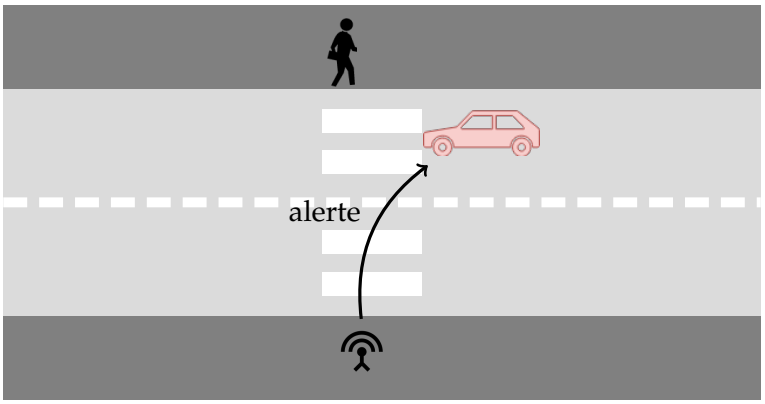


FIGURE 7.4 – Schéma de l'architecture de coopération entre infrastructure et véhicules normée par l'ETSI (cas d'usage D).

Coopération via un serveur central

Il est possible d'éviter à de multiples acteurs de devoir effectuer les calculs permettant de déterminer les risques de collisions en les déléguant à un serveur central. Cela permet de limiter l'usage de ressources sur les extrémités du réseau (véhicules, piétons, infrastructure) et de limiter, dans certains cas, la quantité de calculs à effectuer.

La centralisation de la génération d'alertes facilite généralement la gestion de la confiance en ces alertes, mais il faut tout de même noter que le serveur central se base forcément sur des données issues des autres utilisateurs. Par conséquent, il faut assurer la confiance du serveur central dans les données qui lui sont transmises pour prendre ses décisions.

L'utilisation d'un serveur central éloigne également les calculs de la situation dangereuse, ce qui peut augmenter le temps de réaction global du système de protection et diminuer d'autant son efficacité. Par ailleurs, l'investissement nécessaire au déploiement de ce genre de solution nécessite une infrastructure particulièrement coûteuse (serveur central, unités de bord de route pour la communication avec les véhicules), envisageables uniquement dans certains réseaux comme les autoroutes (où la protection des piétons est un enjeu moins prioritaire).

La littérature propose depuis plusieurs années ce type d'architecture dans [**?**, **?**, **?**]. Cette architecture permet de tirer profit de données en grandes quantités, fournies à la

fois par l'infrastructure et les véhicules, et d'utiliser la fusion de données pour améliorer la confiance dans les conclusions du serveur.

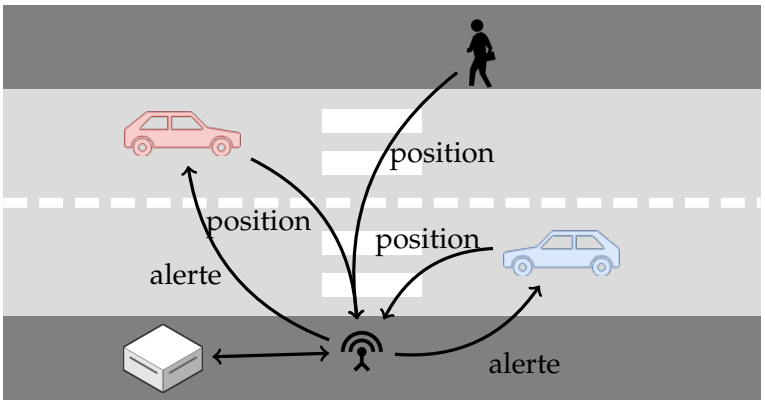


FIGURE 7.5 – Schéma de l'architecture de coopération entre VRU et véhicules via un serveur central, normée par l'ETSI (cas d'usage E).

Coopération entre VRU et véhicule via l'infrastructure

L'infrastructure peut également jouer le rôle de relai (cas d'usage F de la norme [**?**]), assurant la compatibilité entre les technologies de communication utilisées par les véhicules (802.11p, notamment) et celles utilisées par les VRU (wifi, bluetooth, par exemple). Cette approche permet d'augmenter les performances (portée, part des piétons concernés) liées à la coopération entre VRU et véhicules.

Le déploiement d'infrastructure pour jouer le rôle d'intermédiaire entre VRU et véhicules a un coût important qui ne peut être supporté que pour certaines zones particulièrement à risque. En effet, les communications entre l'infrastructure et les VRU auraient une portée très limitée, ce qui oblige à placer de nombreuses bornes de l'infrastructure afin de maintenir la capacité de coopérer avec tous les piétons de la zone.

L'utilisation d'un intermédiaire peut provoquer une latence supplémentaire, diminuant ainsi l'intérêt pour cette solution. Par ailleurs, si l'infrastructure n'agit que comme un intermédiaire, elle ne permet pas d'améliorer la confiance dans les données transmises.

Lorsque la littérature s'intéresse à cette architecture, c'est habituellement en utilisant l'infrastructure des réseaux téléphoniques en tant qu'intermédiaire plutôt que l'infrastructure routière. Cela réduit en effet les coûts de déploiement puisque l'infrastructure existe déjà et simplifie la gestion des compatibilités technologiques. C'est l'approche de [**?**], avec la conception d'une application nommée V2Psense. Elle est basée sur la fonctionnalité de priorité proposée par les réseaux mobiles LTE (Long Term Evolution). Cette application permet d'alerter les piétons et véhicules d'un risque de collision imminente et va donc plus loin que ce que propose la norme ETSI. Dans [**?**], les auteurs proposent une application mobile capable de communiquer avec les véhicules via le réseau mobile. Leur approche originale consiste notamment à utiliser des données personnelles des piétons (notamment leur âge) et une méthode d'apprentissage profond pour affiner la caractérisation du risque de collision.

Dans d'autres travaux comme [**?**], le piéton est intégré au C-ITS grâce à un équipement spécifique nommé PBU (Pedestrian Body Unit), capable de localiser précisément le

piéton. Les auteurs font valoir que les fonctionnalités du PBU peuvent toutes être remplies par un smartphone mais que leurs expériences utilisent un équipement dédié. L'infrastructure et les véhicules peuvent alors calculer les risques de collision et envoyer des alertes aux conducteurs. Cette approche va un peu plus loin que ne le prévoit la norme puisque l'infrastructure peut également générer des alertes à destination de véhicules qui n'auraient pas repéré le risque.

En dehors des applications utilisant le réseau téléphonique, très peu de travaux explorent l'architecture F. L'utilisation du réseau téléphonique peut sembler prometteuse avec le déploiement progressif de la 5G, qui offre des performances intéressantes pour ce type d'usages. Cependant, la couverture réseau pose dans ce cas problème, avec de nombreuses zones blanches, des déconnexions temporaires du réseau téléphonique et l'incapacité, pour le gestionnaire du réseau routier, d'agir sur le réseau téléphonique (qui ne lui appartient pas) en cas de problème.

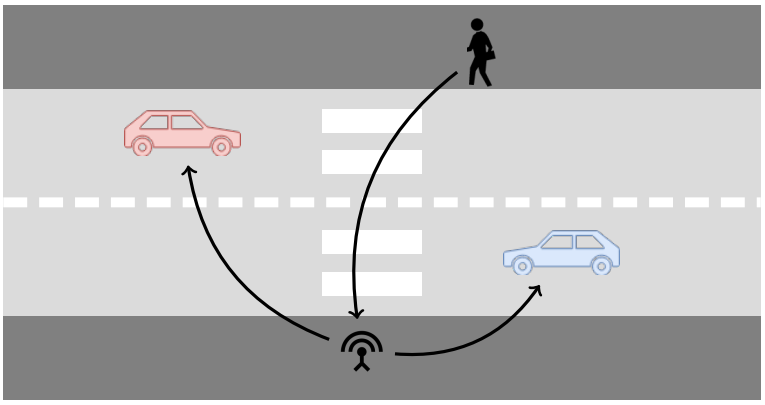


FIGURE 7.6 – Schéma de l'architecture de coopération indirecte entre VRU et véhicules via l'infrastructure, normée par l'ETSI (cas d'usage F).

7.2.3 Discussion

L'ETSI propose une classification des architecture de coopérations pour la protection des VRU dans la norme [**?**] nommées de A à F. Les différentes architectures identifiées par la norme ne font pas de différences entre les VRU, tandis que les travaux de la littérature se restreignent généralement au cas des piétons (profil 1), tandis que le profil 2 (cyclistes et mobilités douces) est plus rarement étudié [**?**]. Le profil 3 (cyclomoteurs) n'est généralement pas considéré dans les travaux sur les VRU.

À l'heure actuelle, toutes les architectures proposées ne sont pas équitablement explorées par la communauté scientifique, notamment parce que les architectures A et C sont presque absentes de la littérature. L'architecture F n'est généralement explorée qu'en utilisant pour intermédiaire le réseau téléphonique, qui pourrait être considéré distinct de l'infrastructure routière.

Une bonne partie des travaux sur les VRU proposent, contrairement aux descriptions de la norme, de faire réagir aussi bien les VRU que le véhicule, même si les détails de l'interface homme machine sur le VRU sont souvent ignorés. Les travaux qui s'y penchent préconisent souvent d'éviter d'utiliser les notifications sur l'écran du téléphone du VRU. Certains proposent d'autres solutions comme [**?**] qui montre que les

alertes sonores produisent des effets positifs sur des piétons qui sont en train d'écrire des SMS. Une approche similaire est proposée par [**?**], avec des notifications visuelles, sonores et des vibrations pour les piétons trop peu attentifs (par exemple écrivant des SMS sur leur téléphone). D'autres travaux préconisent d'alerter les piétons sans utiliser leur téléphone, par exemple grâce à des feux tricolores (qui pourraient être intégrés aux échanges grâce aux unités de l'infrastructure) dans [**?**] ou des avertisseurs visuels et lumineux sur les véhicules dans [**?**, **?**, **?**].

Les architectures A, B, E et F sont basées sur l'émission de messages par les téléphones des VRU. Ces messages sont généralement supposés indiquer la localisation précise du VRU, alors que localiser précisément un téléphone reste une opération complexe, qui consomme beaucoup d'énergie. Différents travaux de recherche comme [**?**] se sont intéressés aux moyens de localiser précisément et sans trop dépenser d'énergie le téléphone d'un piéton. Des stratégies en ce sens sont parfois implémentées dans les solutions de protection des piétons proposées. Malgré des améliorations notables par rapport à un usage simple du GPS, obtenir une précision métrique est relativement complexe et fait souvent appel à une fusion avec les données des accéléromètres. La consommation d'énergie peut être réduite mais semble malgré tout trop importante pour que tous les piétons consentent à utiliser ce type d'applications dès qu'ils sont sur la route.

Plusieurs approches utilisent une solution hybride, mélangeant différents aspects de plusieurs architectures normées. Il est notamment possible de combiner la détection par l'infrastructure proposée dans l'architecture D avec celle par les véhicules en circulation proposée dans l'architecture C. Peuvent également être ajoutées la prise en compte de messages émis par les VRU vers l'infrastructure selon l'architecture F, voire la présence d'un serveur central combinant les infos recueillies par plusieurs bornes de l'infrastructure. Assembler ces différentes approches peut permettre d'utiliser de la fusion de données afin d'augmenter la confiance dans les conclusions de l'analyse.

Afin d'accorder une certaine confiance aux résultats de la coopération, l'architecture D, centralisée et peu sensibles aux données extérieures semble à privilégier. Cependant, combiner cette approche avec de la fusion de données issues des VRU, des véhicules et de leurs capteurs, voire d'autres bornes d'infrastructure semble prometteur pour disposer d'informations plus fiables encore, si cela ne retarde pas trop l'analyse.

Ces éléments issus de la littérature nous permettent désormais de définir le scénario de l'étude expérimentale réalisée.

7.3 Scénario choisi

L'étude par prédiction d'un mécanisme de protection des piétons nécessite le choix d'un scénario dans lequel des piétons sont amenés à se mettre en situation dangereuse, mais aussi d'établir une architecture de coopération pour les protéger et de définir les algorithmes de coopération à utiliser pour éviter la collision.

7.3.1 Configuration routière

Le choix de la configuration routière a une grande influence sur l'étude de cas réalisée. En effet, il impacte à la fois le risque encouru par les VRU et les ressources disponibles pour le prévenir.

La diversité des VRU rend leur inclusion dans un réseau véhiculaire difficile, comme l'a montré la section précédente. En effet, ces utilisateurs de la route ne peuvent pas disposer des mêmes matériels de communication, peuvent avoir des trajectoires radicalement distinctes sur des voies de circulation différentes. Ainsi, une étude complète de la protection des VRU semble trop ambitieuse pour l'étude expérimentale conduite dans ce chapitre.

Les similarités entre les véhicules faiblement motorisés (cyclomoteurs) et les autres véhicules poussent à les considérer comme des véhicules à part entière, dont la vitesse est lente, et donc à les exclure de ce travail.

Le cas d'étude se focalisera, comme beaucoup de travaux de la littérature, sur les piétons, représentatifs de toutes les vulnérabilités des VRU (pas de protection, de source d'énergie, de manœuvrabilité, de règles de circulation strictes ou d'équipements de prévention). Ils représentent en effet à la fois le cas le plus courant et le pire cas parmi les VRU à protéger. Comme il s'agit d'un pire cas, certains aspects de l'étude pourraient être applicables aux autres catégories de VRU mais cela ne fera pas partie des objectifs de l'étude.

Afin d'avoir un impact maximal sur la sécurité routière, la situation routière doit être fréquentée (à la fois par les piétons et par des véhicules). Le danger présenté doit également être important, affecté par exemple par une visibilité limitée et des vitesses véhiculaires suffisantes.

C'est pour ces raisons qu'un scénario urbain, assurant une meilleure fréquentation des piétons est préféré. En ville, les parties des routes les plus fréquentées par des piétons sont généralement matérialisées par un passage-piéton. Ce dernier, même s'il assure une meilleure prise en compte des traversées de piétons par les conducteurs, reste généralement une zone à risque, en particulier s'il n'est pas complété par un système de feux tricolores.

La circulation des véhicules en zone urbaine repose parfois sur des voies rapides destinées à fluidifier la traversée de la ville ou le changement de quartier. Ces voies rapides, semblables à des autoroutes urbaines, disposent souvent d'une limitation de vitesse à 70 km/h, nettement supérieure au reste de la zone urbaine (50 voire 30 km/h). Elles sont cependant également sujettes à la présence d'intersections et de traversées de piétons. Ces caractéristiques en font des zones à privilégier pour la protection coopérative des piétons.

La zone d'intérêt se situera donc sur une autoroute urbaine, au niveau d'une intersection équipée d'un passage-piéton. Les véhicules peuvent rouler assez rapidement à l'approche de cette intersection sur quatre voies séparées par un terre-plein central. Le rond-point matérialisant l'intersection affecte la visibilité du passage piéton et est suffisamment large pour ne pas obliger les véhicules à ralentir beaucoup. La vitesse des véhicules sera ainsi considérée inférieure ou égale à 45 km/h.

7.3.2 Architecture de coopération

À partir du cas d'usage routier choisi, l'architecture de coopération doit être définie. Le choix se portera sur une seule architecture, avec l'objectif de représenter un déploiement le plus réaliste possible, pour représenter les problématiques d'un déploiement réel et faciliter la mise en place d'expériences concrètes.

Puisque la zone choisie est un passage piéton déjà identifié comme dangereux par les autorités, il est intéressant, pour le gestionnaire du réseau routier, d'y installer de l'in-

frastructure. Une telle installation représenterait un investissement raisonnable compte tenu de la dangerosité du site et de sa fréquentation. On prend donc comme hypothèse de travail la présence d'infrastructure routière communiquante au niveau de l'intersection.

Afin d'assurer la prise en compte des piétons non équipés, les piétons doivent être détectés par un capteur qui leur est externe. Du fait de sa présence supposée sur l'intersection considérée, c'est l'infrastructure qui est chargée de détecter et signaler la présence de piétons. L'architecture de coopération choisie correspond ainsi au cas d'usage D de la norme ETSI [**?**], comme le montre la figure 7.7.

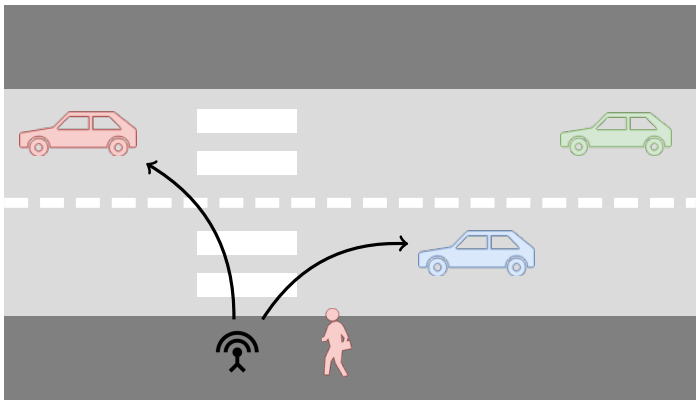


FIGURE 7.7 – Schéma du scénario étudié, avec l'architecture de coopération depuis l'infrastructure vers les véhicules.

7.3.3 Algorithmes de protection

Différents algorithmes simples seront testés pour permettre l'étude de plusieurs mécanismes différents. Ces algorithmes sont malgré tout proches des déploiements réellement envisagés, leur simplicité découlant plutôt du nombre limité d'acteurs impliqués à chaque traversée.

Algorithme *Alert push*

La première possibilité est qu'une borne de l'infrastructure diffuse régulièrement (approche *push*, où la borne pousse les alertes vers les véhicules) l'état du passage piéton, s'il est occupé et présente donc un danger. Lorsqu'un véhicule approche de la borne pendant qu'un piéton traverse, il commence à recevoir les alertes de la part de la borne de bord de route. Il peut alors adapter son allure pour faciliter un éventuel arrêt avant d'être en vue de la zone dangereuse.

La borne dispose d'une primitive TRAVERSEE qui retourne vrai si le passage piéton est occupé lors de son appel, et faux sinon. Ce comportement est décrit plus en détails à l'algorithme 9.

Algorithme 9 : **Alert push**

1 Initialisation :
  ▷ *exécutée uniquement sur la borne de l'infrastructure*
2 Armer le timer

3 Expiration du timer :
4 **si** TRAVERSEE() **alors**
5 **envoyer(** ALERT)
6 **fin si**
7 Réarmer le timer

8 Réception d'un message ALERT sur un véhicule :
9 Déclenchement de l'alerte conducteur

Cet algorithme simple suffit à alerter les véhicules proches du passage piéton lorsqu'il y a un danger.

Algorithme *Alert pull*

Les applications de C-ITS supposent habituellement des messages périodiques émis par les véhicules. Ce type de comportement est également celui prescrit par la plupart des standards (notamment [**?**, **?**]). Lorsque l'infrastructure détecte un véhicule via ses émissions périodiques, alors qu'une traversée est en cours, elle lui répond par une alerte. Les alertes sont ainsi transmises aux véhicules par une approche *pull* (où les véhicules tirent les alertes à eux), à l'intérieur de la réponse de l'infrastructure. Cet algorithme vise à limiter les ressources utilisées par la borne, notamment l'énergie lorsqu'aucun véhicule n'approche, mais également les ressources réseau lorsque peu de véhicules sont en circulation. Elle est décrite à l'algorithme 10.

Algorithme 10 : **Alert pull**

1 Initialisation :
2 *date_dernier_envoi* $\leftarrow$ date_actuelle()

3 Réception d'une balise véhiculaire :
4 **recevoir(** BALISE)
5 **si** date_actuelle() $-$ *dernier_envoi* $>$ *DELAI_INTER_ALERTES* **alors**
6 **si** TRAVERSEE() **alors**
7 **envoyer(** ALERT)
8 *date_dernier_envoi* $\leftarrow$ date_actuelle()
9 **fin si**
10 **fin si**

L'utilisation de ressources réseau par l'infrastructure est limitée grâce à l'utilisation du délai *DELAI_INTER_ALERTES*, qui représente la durée minimale entre deux envois de l'infrastructure.

Algorithme Diffusion d'alertes à *n* sauts

Retransmettre une alerte sur plusieurs sauts peut permettre à la fois à des véhicules hors de portée de mieux anticiper le danger et aux véhicules à portée de bénéficier de transmissions multiples, qui peuvent être utiles pour pallier les pertes aléatoires. L'algorithme de diffusion dispose d'un paramètre noté n représentant la profondeur maximale

atteinte par les alertes (le nombre maximal de sauts entre le RSU source de l'alerte et le véhicule qui la reçoit).

Algorithme 11 : **Diffusion d'alerte à *n* sauts**

```
1   Initialisation :
      ▷ Numéro de séquence d'une alerte
2     seq ← 0
      ▷ Liste des numéros de séquence des alertes reçues
3     historique ← ∅
      ▷ exécuté uniquement sur la borne de l'infrastructure
4     Armer le timer

5   Expiration du timer :
6     si TRAVERSEE() alors
7       seq ← seq + 1
8       envoyer( ALERT, numero = seq, niveau = n )
9     fin si
10    Réarmer le timer

11  Réception d'une alerte :
12    recevoir( ALERTE, numero = s, niveau = n )
13    si n > 1 et s ∉ historique alors
14      envoyer( ALERT, numero = seq, niveau = n − 1 )
15    fin si
```

La transmission des alertes permet de prévenir des véhicules éloignés qui ne sont pas à portée de communication du RSU qui les génère. Ce procédé permet notamment de limiter la fréquence d'envoi (la surface couverte par une émission d'alerte est plus grande), donc l'usage de ressources réseau par le RSU. Pour l'implémentation de cet algorithme, afin d'éviter les dépassements de mémoire, il est possible de compresser la liste représentant l'historique des alertes reçues *historique*, par exemple en la remplaçant par une liste d'intervalles, ou en affectant une date d'expiration à chaque alerte, à partir de laquelle elle est supprimée de l'historique.

7.3.4 Prédiction

La prédiction du résultat de l'un des algorithmes de coopération choisis est effectuée selon la méthode proposée au chapitre 3. Elle est réalisée grâce à une analyse du fonctionnement de l'algorithme réparti et a pour résultat un prédicat appelé propriété de performance topologique. Ce dernier ne dépend que de la topologie dynamique observée, et peut donc être analysé en l'absence d'exécution de l'application de coopération véhiculaire.

Choix des hypothèses

Dans la réalité, des paramètres difficiles à prévoir (non liés à la position relative des nœuds du réseau) peuvent influencer les transmissions et les faire échouer. Ces pertes aléatoires doivent être prises en compte pour effectuer une prédiction réaliste. Du fait de

leur nature difficile à prévoir elles doivent être prises en compte au cours de l'analyse algorithmique dans l'élaboration de la propriété de performance topologique.

Dans la modélisation du scénario, on considère que chaque émission a une probabilité notée τ de ne pas être reçue par son récepteur. Cette probabilité est supposée constante pour simplifier l'analyse et chaque réception (même lorsque deux réceptions sont issues de la même émission) est considérée comme un événement aléatoire indépendant des autres. Ainsi, les prédictions réalisées dans le chapitre 6 constituent un cas particulier de cette modélisation dans laquelle toutes les réceptions ont une probabilité $\tau = 0\,\%$ d'être perdues.

La norme ETSI [**?**] préconise que le conducteur dispose, à réception de l'alerte, d'un temps restant à collision (TTC pour *Time To Collision*) d'au moins 5 s pour éviter la collision sans provoquer d'autres mises en danger. Ainsi, pour chaque algorithme de coopération testé, la propriété de performance algorithmique sera la bonne réception d'au moins 1 message d'alerte avec un TTC d'au moins 5 s.

La possibilité de pertes de messages survenant aléatoirement ($\tau > 0$) rend toute prédiction déterministe impossible. En effet, même si les conditions topologiques sont très bonnes, il reste possible que des pertes surviennent, par hasard à chaque transmission de message, ce qui empêcherait le succès de l'algorithme. C'est pourquoi la propriété de performance algorithmique sera la bonne réception d'au moins 1 message d'alerte avec un TTC d'au moins 5 s avec une probabilité de 90 %.

Propriété de performance topologique avec *Alert push*

Le fonctionnement de l'algorithme *Alert push* repose sur l'émission régulière, lorsqu'il y a un danger, d'un message d'alerte par l'infrastructure. La réaction du véhicule à risque de collision ne nécessite qu'une unique réception du message d'alerte, pour simplifier les questions de gestion de la confiance.

Lorsqu'aucune perte aléatoire n'a lieu, la propriété de performance topologique peut être l'existence, dans le 1-graphe dynamique représentant l'observation (jusqu'à une date t_f antérieure de plus de 5 s à la collision sans réaction du conducteur $t_{collision}$), d'une arête reliant la borne d'infrastructure et le véhicule.

Lorsque le taux de pertes est τ, la présence d'une p-arête garantit qu'un message périodique d'alerte sera reçu avec une probabilité $1 - \tau^p$. Ainsi, la présence, dans le p-graphe dynamique représentant le scénario jusqu'à la date $t_f = t_{collision} - 5$, d'une arête reliant le véhicule et la borne de l'infrastructure garantit que le message d'alerte sera reçu par le véhicule avec une probabilité de $1 - \tau^p$ si la p-arête est postérieure au démarrage de l'alerte sur la borne t_{alerte}.

La propriété de performance topologique est donc la présence d'une p-arête reliant la borne et le véhicule dans le p-graphe dynamique représentant l'observation entre les dates t_{alerte} et t_f. Le choix de la valeur de p est ainsi fonction du taux de pertes de l'essai, selon l'équation 7.1.

$$1 - \tau^p \geq 0,90 \tag{7.1}$$

Propriété de performance topologique avec *Alert pull*

Le fonctionnement de l'algorithme *Alert pull* repose sur la réception par l'infrastructure d'une balise émise par un véhicule approchant. Lors de cette réception, elle émet

une alerte si la situation présente un danger et cette alerte, pour être efficace, doit également être reçue par le véhicule à l'approche.

Lorsqu'aucune perte aléatoire n'a lieu, la propriété de performance topologique peut être l'existence, dans le 2-graphe dynamique représentant l'observation jusqu'à la date $t_f = t_{collision} - 5$), d'une arête reliant la borne d'infrastructure et le véhicule. Cette arête permettrait l'échange de 2 messages consécutifs, l'un étant une balise périodique du véhicule (message CAM, par exemple) et l'autre l'alerte émise en réponse par la borne.

Lorsque le taux de pertes est τ, la présence d'une 2-arête garantit que le premier message périodique du véhicule sera reçu avec une probabilité $1 - \tau$ par la borne. Si elle reçoit bien ce message, elle renvoie une alerte qui aura également une probabilité $1 - \tau$ d'être reçue par le véhicule. La probabilité, en présence d'une 2-arête, que le véhicule reçoive l'alerte est alors de $p_{succès} = (1-\tau)^2$. Puisque cet échange peut avoir lieu à chaque émission périodique du véhicule, la probabilité que l'alerte ne soit pas reçue au cours de l'émission (à portée adéquate) de n balises périodiques par le véhicules est alors donnée par la loi binomiale : $\mathbb{P}(X = 0) = \binom{n}{0} \times p_{succès}^0 \times (1 - p_{succès})^n$. La probabilité que l'alerte soit reçue est alors : $\mathbb{P}(X \geq 1) = 1 - \mathbb{P}(X = 0) = 1 - (1 - p_{succès})^n = 1 - (1 - (1 - \tau)^2)^n$

En présence d'une p-arête, le véhicule émet p balises, et, en l'absence de pertes, la borne renvoie $p - 1$ alertes. Ainsi, la présence, dans le p-graphe dynamique représentant le scénario à une date t, d'une arête reliant le véhicule et la borne de l'infrastructure garantit que le message d'alerte sera reçu par le véhicule avec une probabilité de $1 - \big((1 - \tau)(1 - (1 - \tau)^2)^{p-1}\big)$ si la p-arête est postérieure au démarrage de l'alerte sur la borne t_{alerte} et antérieure d'au moins 5 s à la date de collision sans réaction du conducteur.

La propriété de performance topologique est donc la présence, dans le p-graphe dynamique représentant l'observation entre la date t_{alerte} et la date t_f d'une arête reliant la borne et le véhicule. La valeur de p est alors fonction du taux de pertes τ selon l'équation 7.2.

$$1 - (1 - (1 - \tau)^2)^{p-1} \geq 0,9 \tag{7.2}$$

Propriété de performance topologique avec Diffusion d'alerte à *n* sauts

Le fonctionnement de l'algorithme de diffusion d'alerte repose sur l'émission régulière, lorsqu'il y a un danger, d'un message d'alerte par l'infrastructure.

Lorsqu'aucune perte n'a lieu, la propriété de performance topologique est simplement l'existence, dans un p-graphe dynamique, d'un chemin de longueur telle que $l \leq n$ et $l \leq p$ reliant le RSU à chacun des véhicules au moins 5 s avant la collision virtuelle entre ce véhicule et le piéton.

Lorsque le taux de pertes est τ, un message doit, pour pouvoir être retransmis, être reçu par chaque nœud intermédiaire du chemin parcouru. Chaque transmission ayant une probabilité τ d'échouer, la probabilité que la transmission parcourt tout le chemin est donc $p_{succès} = (1 - \tau)^l$ où l est la longueur du chemin. Si p messages sont émis à l'origine du chemin, la probabilité qu'aucun message ne soit reçu est donc obtenue par la loi binomiale $\mathbb{P}(X = 0) = \binom{n}{0} \times p_{succès}^0 \times (1 - p_{succès})^p = (1 - p_{succès})^p$. La probabilité que l'alerte soit reçue par ce chemin est alors donnée par l'équation 7.3.

$$\mathbb{P}(X \geq 1) = 1 - \mathbb{P}(X = 0) = 1 - (1 - p_{succès})^p = 1 - (1 - (1 - \tau)^l)^p \tag{7.3}$$

7.4 Étude expérimentale

Afin de mettre à l'épreuve les prédictions réalisées, un protocole expérimental est établi, et permet dans chaque variante du scénario (utilisant un algorithme de coopération différent) de confronter la prédiction à la réalité.

7.4.1 Protocole expérimental

Les expérimentations sont réalisées en émulateur, en considérant des véhicules sur des trajectoires entre 30 et 45 km/h à l'approche du carrefour. L'alerte démarre toujours avant que le premier véhicule n'entre à portée de la borne et est continue jusqu'à ce que le dernier véhicule ait dépassé le carrefour. Le comportement des véhicules à réception de l'alerte n'est pas modifié, afin de déterminer le temps restant à collision (TTC).

Chaque expérience est menée avec un ou plusieurs véhicules ayant un point de départ aléatoire et une vitesse aléatoire (parmi 30,35,40,45 km/h) avant le carrefour. Les communications entre l'infrastructure et les véhicules sont considérées possibles uniquement à une portée de 300 m, et peuvent aléatoirement donner lieu à des pertes de messages.

Le taux de pertes est défini pour chaque expérience entre $\tau = 0\,\%$ et $\tau = 70\,\%$. Cette valeur est constante au cours du temps et identique sur tous les nœuds. On calcule, pour chaque taux de pertes testé, la probabilité de réception de l'alerte. De manière similaire aux expériences de validation proposées au chapitre 4, on définit 4 issues possibles décrivant la qualité des prédictions :

— succès prédit, si l'alerte est reçue avec un TTC d'au moins 5 s et que la propriété de performance topologique est remplie,

— échec prédit, si l'alerte n'est pas reçue à temps et que la propriété de performance topologique n'est pas remplie,

— succès non prédit, si l'alerte est reçue avec un TTC d'au moins 5 s alors que la propriété de performance topologique n'est pas remplie,

— échec non prédit, si l'alerte n'est pas reçue à temps alors que la propriété de performance topologique est remplie.

Dans la mesure où chaque réception est aléatoire, des échecs non prédits vont forcément survenir par manque de chance. S'ils ne dépassent pas 10 % des tests, ils ne remettent pas en cause la prédiction (puisque la propriété de performance algorithmique est que l'alerte soit reçue à temps dans 90 % des cas où on le prédit).

7.4.2 Résultats avec *Alert push*

Les expériences concernant l'algorithme *Alert push* sont réalisées en émulateur, avec 3 véhicules lors de chaque test. Ils n'interagissent jamais entre eux et n'émettent jamais de messages, mais reçoivent ceux de la borne de l'infrastructure. Puisque différents taux de pertes sont étudiés, la valeur de p pour satisfaire la propriété de performance topologique est calculée pour chacun d'entre eux. La table 7.1 résume les valeurs obtenues.

Sur chaque test est isolée l'observation jusqu'à la date t_f. Elle est utilisée pour calculer la famille de p-graphes dynamiques grâce à laquelle est effectuée une prédiction, qui est comparée à la date de première réception de l'alerte. On obtient ainsi la qualité de la

Taux de pertes (%)	0	10	20	30	40	50	60	70
Valeur de p	1	1	2	2	3	4	5	7
Probabilité de réception (%)	100	90	96	91	94	94	93	92

TABLE 7.1 – Propriétés de performance topologique en fonction du taux de pertes (algorithme *Alert push*). La présence de la p-arête recherchée assure une probabilité supérieure à 90 % de recevoir l'alerte dans les temps

Taux de pertes (%)	0	10	20	30	40	50	60	70
Valeur de p	2	3	4	5	6	9	14	25
Probabilité de réception (%)	100	97	96	94	90	90	90	90

TABLE 7.2 – Propriétés de performance topologique en fonction du taux de pertes (algorithme *Alert pull*). La présence de la p-arête recherchée assure une probabilité supérieure à 90 % de recevoir l'alerte dans les temps

prédiction pour chaque test. Les qualités des prédictions réalisées sont présentées à la figure 7.8 selon la même présentation que dans le chapitre 6. .

La figure -a montre que les expériences qui se sont déroulées avec un faible taux de pertes (jusqu'à 30 %) ont des résultats similaires. Environ 3 tests sur 4 assurent une réception à temps de l'alerte (cette valeur baisse un peu lorsque le taux de pertes augmente), et 90 % d'entre eux sont correctement prédits. Les tests restants ne permettent pas une réception de l'alerte 5 s avant la collision virtuelle et la proportion d'échecs parmi eux est négligeable.

La figure -b montre que les expériences qui se sont déroulées avec un taux de pertes plus fort (de 30 à 70 %) montrent que la proportion de succès parmi les tests diminue lorsque le taux de pertes augmente (passant de 75 % pour un taux de pertes de 40 % à 60 % pour un taux de pertes de 70 %). La proportion de succès non prédits passe de 10 % des succès avec un taux de pertes de 40 % à 20 % des succès avec un taux de pertes de 70 %, ce qui représente un peu plus de 10 % des tests. La proportion des échecs non prédits suit une dynamique semblable, et représente toujours moins de 10 % des tests.

Ces résultats montrent que la méthode proposée au chapitre 5 permettrait de valider l'utilisation de l'algorithme *Alert push* pour alerter les véhicules de la présence d'un piéton dans un contexte routier réaliste.

7.4.3 Résultats avec *Alert pull*

Les expériences concernant l'algorithme *Alert pull* sont réalisées en émulateur, avec 1 véhicule lors de chaque test. Il n'interagit qu'avec la borne de l'infrastructure. Avec cet algorithme, c'est le véhicule qui émet des messages périodiques et la borne qui leur répond.

Puisque différents taux de pertes sont étudiés, la valeur de p dans la propriété de performance topologique est calculée pour chacun d'entre eux. La table 7.2 résume les valeurs obtenues.

Sur chaque test est effectuée une prédiction, qui est comparée à la date de première réception de l'alerte. On obtient ainsi la qualité de la prédiction pour chaque test. Les qualités des prédictions réalisées sont présentées à la figure 7.9 selon la même présentation que dans le chapitre 6.

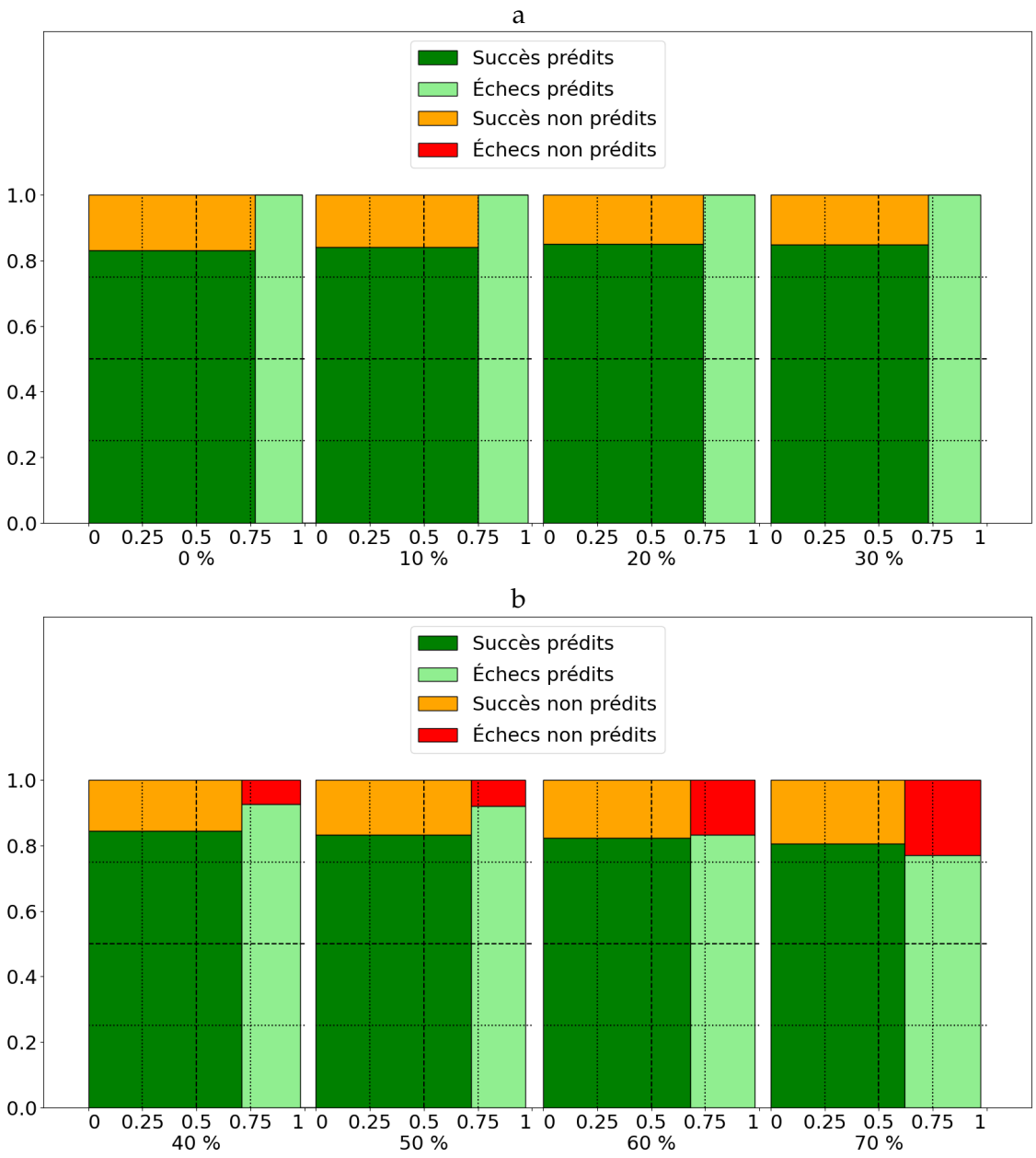


FIGURE 7.8 – Qualité des prédictions sur les expériences impliquant l'algorithme *Alert push* selon le taux de pertes programmé dans l'essai. Sur le graphique a, les taux de pertes les plus bas donnent lieu à un nombre négligeable d'échecs non prédits. Sur le graphique b, les taux de pertes les plus élevés donnent lieu à une proportion acceptable (inférieure à 10 %) d'échecs non prédits.

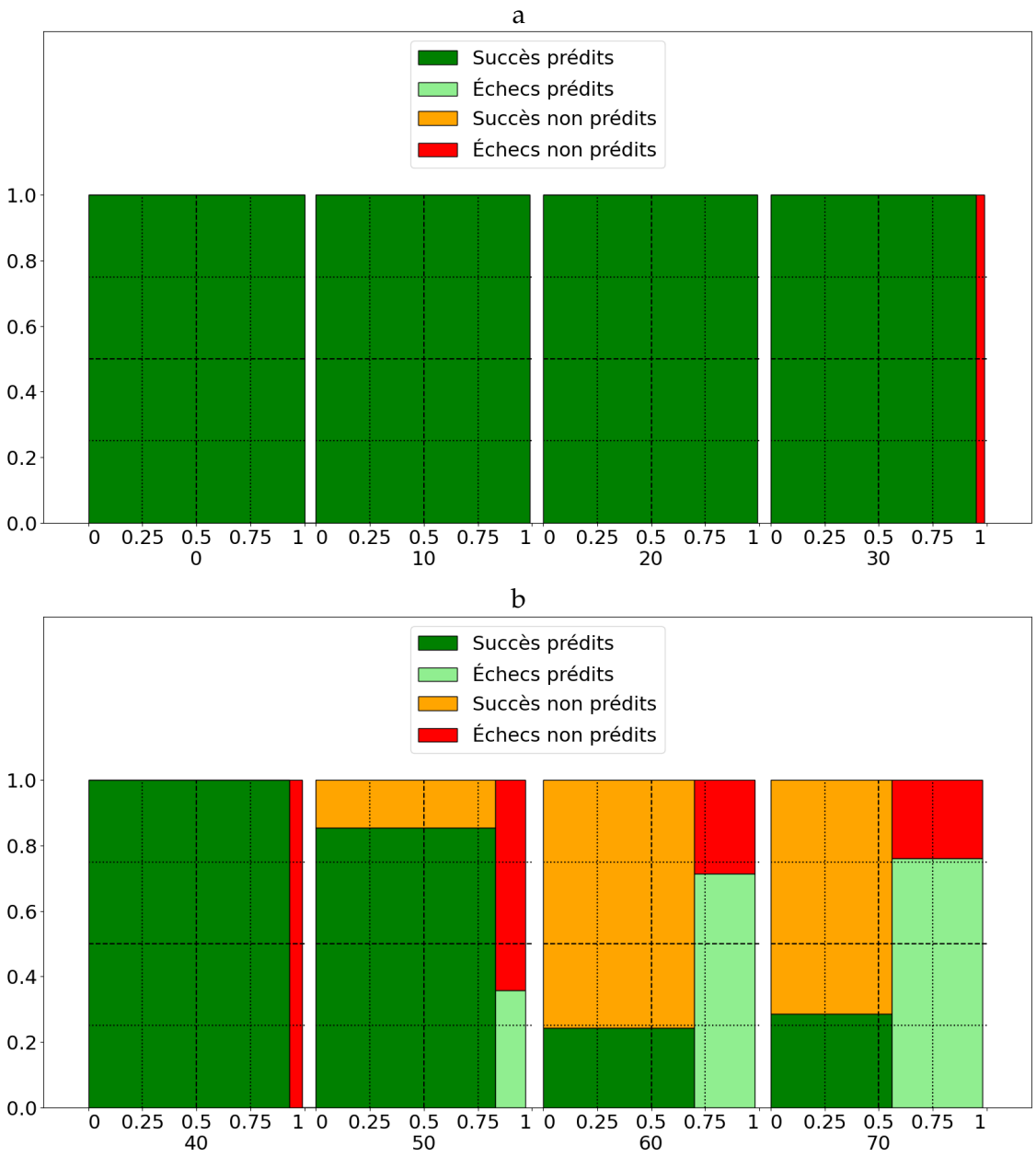


FIGURE 7.9 – Qualité des prédictions sur les expériences impliquant l'algorithme *Alert pull* selon le taux de pertes programmé dans l'essai. Sur le graphique a, les taux de pertes les plus bas donnent lieu à un nombre négligeable d'échecs, généralement non prédits. Sur le graphique b, lorsque le taux de pertes augmente, des échecs non prédits apparaissent, mais leur proportion reste acceptable (inférieure à 10 %).

Taux de pertes (%)	0	10	20	30	40	50	60	70
Valeur de p	1	1	2	2	3	4	5	7
Probabilité de réception (%)	100	90	96	91	94	94	93	92

TABLE 7.3 – Propriétés de performance topologique en fonction du taux de pertes (algorithme Diffusion d'alertes à 2 sauts). La présence de la p-arête recherchée assure une probabilité supérieure à 90 % de recevoir l'alerte dans les temps s'il existe un chemin direct entre le RSU et le véhicule.

La figure -a montre que les expériences qui se sont déroulées avec un faible taux de pertes (jusqu'à 30 %) ont des résultats similaires, avec une proportion négligeable d'échecs algorithmiques. Certains d'entre eux sont non prédits, en raison de la nature aléatoire de la propriété de performance algorithmique (chaque véhicule a 90 % de chances de recevoir à temps l'alerte). Leur proportion reste cependant très en deça de 10 % des tests, ce qui signifie qu'elle est acceptable au vu de la propriété de performance algorithmique.

La figure -b montre que les expériences réalisées avec un taux de pertes plus élevé (40 à 70 %) donnent lieu à une proportion d'échecs algorithmiques plus importante, et croissante avec le taux de pertes. La proportion des échecs non prédits, quant à elle, reste contenue en deça des 10 % prévus par la propriété de performance algorithmique.

La méthode de prédiction proposée au chapitre 5 s'est ainsi montrée capable de prédire les conditions de bon fonctionnement pour l'algorithme *Alert pull*.

7.4.4 Résultats avec Diffusion d'alertes à 2 sauts

Les expériences concernant l'algorithme Diffusion d'alerte à 2 sauts sont réalisées en émulateur, avec 2 véhicules lors de chaque test. Ils interagissent avec la borne de l'infrastructure et entre eux pour la transmission d'alerte

Puisque différents taux de pertes sont étudiés, la valeur de p dans la propriété de performance topologique est calculée pour chacun d'entre eux. La table 7.2 résume les valeurs obtenues.

Si on considère une profondeur maximale de transmission de 2 sauts, la propriété de performance topologique est alors la présence dans le p-graphe, d'un chemin du RSU au véhicule, avec la valeur de p telle qu'exprimée à l'équation 7.4.

La valeur de p est dépendante du taux de pertes τ, et des chemins disponibles. Dans la mesure où il n'y a que deux véhicules, elle peut être résumée par l'équation 7.4 où $\text{directe}() = 1 - (1 - (1 - \tau))^p$ lorsqu'il y a un chemin direct et $\text{directe}() = 0$ sinon, et $\text{indirecte}() = 1 - (1 - (1 - \tau)^2)^p$ s'il y a un chemin indirect et $\text{indirecte}() = 0$ sinon.

$$\text{directe}() + \text{indirecte}() - \text{directe}() \times \text{indirecte}() \geq 0,90 \tag{7.4}$$

La valeur de p peut ainsi être calculée selon les situations (le taux de pertes τ et les chemins disponibles entre le RSU et le véhicule). La table 7.3 indique les valeurs de p lorsqu'un chemin direct relie le véhicule et le RSU, tandis que la table 7.4 montre ces mêmes valeurs dans le cas d'un chemin indirect. Lorsque les deux chemins sont disponibles, les valeurs de p en fonction du taux de pertes sont indiquées à la table 7.5.

Sur chaque test est effectuée une prédiction, qui est comparée à la date de première réception de l'alerte. On obtient ainsi la qualité de la prédiction pour chaque test. Les

Taux de pertes (%)	0	10	20	30	40	50	60	70
Valeur de p	1	2	3	4	5	8	13	24
Probabilité de réception (%)	100	97	96	94	90	90	90	90

TABLE 7.4 – Propriétés de performance topologique en fonction du taux de pertes (algorithme Diffusion d'alertes à 2 sauts). La présence de la p-arête recherchée assure une probabilité supérieure à 90 % de recevoir l'alerte dans les temps s'il existe un chemin indirect entre le RSU et le véhicule.

Taux de pertes (%)	0	10	20	30	40	50	60	70
Valeur de p	1	1	1	2	2	3	4	5
Probabilité de réception (%)	100	99	93	98	94	95	95	91

TABLE 7.5 – Propriétés de performance topologique en fonction du taux de pertes (algorithme *Alert pull*). La présence de la p-arête recherchée assure une probabilité supérieure à 90 % de recevoir l'alerte dans les temps s'il existe un chemin direct et un chemin indirect entre le RSU et le véhicule.

qualités des prédictions réalisées sont présentées à la figure 7.10 selon la même présentation que dans le chapitre 6.

La figure -a montre que les expériences réalisées avec un faible taux de pertes (jusqu'à 30 %) ont des résultats similaires, avec une faible proportion d'échecs algorithmiques, systématiquement bien prédits. La figure -b montre également des résultats similaires lorsque le taux de pertes augmente, avec une part croissante de prédictions d'échecs. Des succès non prédits surviennent cependant dans tous les scénarios, même en l'absence complète de pertes. Cela peut être lié aux approximations de modélisation (la fonction de durée de transfert δ est choisie proportionnelle alors qu'il est techniquement possible de transmettre un message en une durée inférieure à $\delta(1)$) et n'affecte pas négativement l'utilité des prédiction en termes de sécurité routière.

La proportion d'échecs non prédits est négligeable, nettement en deça des 10 % prévus par la propriété de performance algorithmique. La méthode de prédiction est ainsi capable de déterminer les conditions de fonctionnement nominal de l'algorithme de Diffusion d'alerte et d'une application qui l'implémente.

7.5 Conclusion

L'étude d'un cas concret de sécurité routière a permis d'illustrer l'utilisation de la méthode de prédiction présentée au chapitre 5 dans des circonstances réalistes. Elle a ainsi montré que des résultats similaires aux situations plus théoriques évoquées au chapitre 6 peuvent être obtenus dans des situations plus réalistes. L'une des nouveautés apportées par ce cas d'étude est qu'il a permis de prendre en compte les pertes de messages non liées à la topologie du réseau, au sein de l'étude formelle permettant d'élaborer la propriété de performance topologique.

Le scénario étudié montre un besoin réel de solutions de coopération, auquel la littérature scientifique s'est particulièrement intéressée ces dernières années. Les solutions envisagées ont un fort impact sur la sécurité des piétons et doivent donc disposer de validations particulièrement concrètes et convainquantes, comme c'est le cas de la méthode

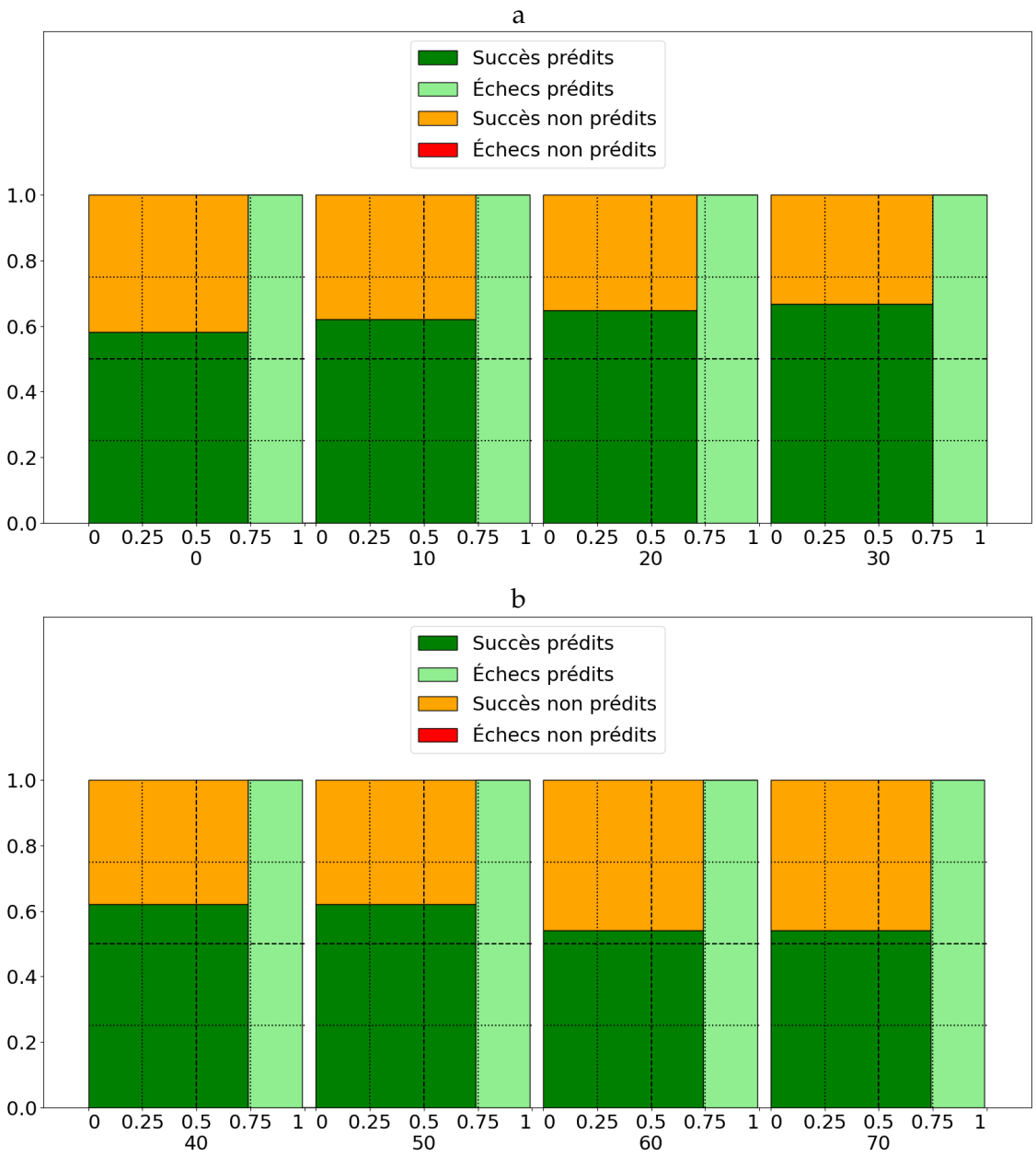


FIGURE 7.10 – Qualité des prédictions sur les expériences impliquant l'algorithme Diffusion d'alertes selon le taux de pertes programmé dans l'essai. Sur le graphique a, les taux de pertes les plus bas donnent lieu à un nombre négligeable d'échecs, généralement non prédits. Sur le graphique b, lorsque le taux de pertes augmente, des échecs non prédits apparaissent, mais leur proportion reste acceptable (inférieure à 10 %).

proposée.

Des expériences réelles sur routes sont prévues pour assurer un réalisme accru des résultats. Elles permettront de réaliser une preuve de concept de déploiement d'une application véhiculaire coopérative à l'aide de la prédiction.

L'utilisation du modèle des p-graphes dynamiques pour exprimer des propriétés topologiques ayant un certain réalisme en réseau de véhicules semble également prometteuse pour réaliser des preuves plus formelles d'algorithmes véhiculaires.

Chapitre 8

Conclusion et perspectives

8.1 Conclusion générale

Les Systèmes de Transports Intelligents Coopératifs (C-ITS) constituent une avancée technologique majeure qui a un fort potentiel pour l'avenir de la mobilité. Le déploiement de ces systèmes vise à réduire les accidents de la route, à gérer efficacement le trafic et à améliorer l'expérience de conduite tout en réduisant l'impact écologique. Les services C-ITS impliquent une interaction entre les véhicules connectés et leur environnement.

Le réseau de communication joue un rôle clé dans les applications véhiculaires coopératives. Il est un élément fondamental qui permet une communication efficace entre les véhicules et les infrastructures qui les entourent. Néanmoins, les caractéristiques du réseau ad hoc véhiculaire (VANET) posent des défis spécifiques à la conception et au déploiement d'applications véhiculaires. En effet, l'absence de routage global et d'adressage fixe, ainsi que de l'hétérogénéité du matériel et des logiciels utilisés par les acteurs du réseau routier entravent fortement leur coopération. De plus, la forte mobilité des véhicules (même si elle est partiellement prévisible), les obstacles fixes et mobiles présents dans l'environnement peuvent affecter la qualité de la communication. Ces contraintes rendent complexes et coûteux la conception et le déploiement expérimental de solutions C-ITS, ce qui limite les capacités de validation de ces solutions et ralentit leur déploiement réel.

Les travaux de cette thèse sont centrés sur la validation d'applications de coopération véhiculaire. Des méthodes empiriques et théoriques de validation sont utilisées pour en cerner les limites. Nous avons tout d'abord étudié les réseaux de coopération entre véhicules, ce qui a permis, au cours du chapitre 2, de cerner les contours des travaux de thèse et les difficultés à surmonter avant de poser des hypothèses de travail.

La première étape de la contribution a consisté en la conception et à la validation empirique d'un algorithme de diffusion fiable en réseau véhiculaire nommé RDF. Cet algorithme, décrit dans le chapitre 3, se distingue notamment par la prise en compte de ressources mémoire et réseau limitées caractéristiques des VANET et du matériel embarqué dans les véhicules. La validation empirique d'une implémentation de cet algorithme s'est effectuée à la fois sur une expérience concrète de preuve de concept et sur des expériences émulées permettant d'en caractériser les performances.

Face aux insuffisances des validations expérimentales menées au chapitre 3, une alternative consiste à réaliser une étude formelle. Cette dernière nécessite des modélisa-

tions de la topologie réseau qui permettent de prendre en compte la grande mobilité des véhicules. Nous proposons alors une étude des modélisations permettant de représenter fidèlement un réseau de topologie dynamique au chapitre 4. Le modèle des p-graphes dynamiques semble adapté à l'expression de propriétés décrivant la topologie d'un réseau dynamique de véhicules et sera utilisé par la suite.

Nous avons alors pu proposer une approche de validation d'algorithmes intermédiaire, entre une validation expérimentale et une validation théorique. Cette méthode, décrite dans le chapitre 5, consiste à effectuer une prédiction sur le comportement de l'algorithme de coopération véhiculaire dans différentes topologies de réseaux dynamiques, représentatives du cas d'application envisagé. La prédiction ainsi réalisée, qui ne dépend que de la topologie réseau a le potentiel d'offrir des garanties suffisantes dans le cadre d'un déploiement d'application coopérative de sécurité routière. La première étape de la prédiction consiste en l'étude de l'algorithme implémenté par l'application de coopération. Cette étude permet de faire le lien entre le fonctionnement de l'algorithme et les propriétés topologiques du réseau assurant son succès. L'intérêt de la méthode proposée est qu'elle ne nécessite aucun travail d'implémentation de l'application de coopération si son algorithme est connu. La prédiction se base cependant sur l'analyse d'observations de réseau dynamique, ce qui lui permet d'assurer un certain degré de réalisme. Elle n'est en revanche applicable que si l'on dispose d'observations suffisamment représentatives de la situation dans laquelle on souhaite déployer l'application.

Une évaluation de la méthode de prédiction est alors proposée. Elle utilise des applications simples de coopération véhiculaire. Cette évaluation démontre la valeur prédictive de la méthode et sa capacité à déterminer si un scénario routier permet le bon fonctionnement de l'application. Elle permet notamment de montrer comment les approximations effectuées dans l'analyse algorithmique affectent l'exactitude des prédictions réalisées. L'évaluation de la méthode proposée a permis de s'assurer qu'il est possible de prédire correctement tous les échecs algorithmiques lorsque les conditions choisies par l'étude algorithmique sont des conditions suffisantes pour remplir les spécifications les spécifications de l'algorithme. De telles prédictions permettent d'assurer le bon fonctionnement de l'algorithme, donc d'assurer certaines propriétés de sécurité routière à condition que la prédiction soit réalisée sur des observations représentatives de la situation réelle. En particulier, les tests d'évaluation ont été conduits en considérant un medium de transmission des messages parfait, dans lequel aucune perte de message n'a lieu hormis celles résultant de la mobilité des véhicules.

Une dernière contribution consiste à appliquer la méthode de prédiction à un cas concret d'application de coopération véhiculaire pour la sécurité routière. Cette application consiste à prévenir les collisions entre véhicules et piétons en alertant les véhicules des risques de collision à l'avance. Cette étude a utilisé des stratégies de protection des piétons issues de la littérature scientifique afin d'améliorer les connaissances des véhicules sur un carrefour urbain dangereux. Grâce à l'utilisation de ce cas d'application concret, il a été possible de prendre en compte un medium de communication imparfait, pouvant provoquer des pertes de messages. Même dans ces conditions, la méthode de prédiction proposée a montré son efficacité pour prédire les performances des applications de protection des piétons.

8.2 Perspectives

Les réseaux véhiculaires constituent un domaine de recherche ouvert, qui se base sur des technologies de plus en plus performantes. La prédiction de performances d'applications de coopération véhiculaire via la méthode proposée dans cette thèse doit toujours être validée en conditions réelles, à travers des expériences routières. De telles expériences sont prévues dans la suite de notre travail, en reprenant l'architecture de protection des piétons étudiée au chapitre 7. Elles permettront à la fois de vérifier l'applicabilité des prédictions (démonstration conceptuelle) et de tester la prise en compte de paramètres réels mesurés comme le taux de pertes, la portée du système de communication.

L'étude de scénarios de sécurité routière plus complexes est également envisagée à plus long terme afin de tester la méthode dans des conditions impliquant plus de véhicules ayant des interactions plus variées, comme c'est envisagé dans le cadre de la ville intelligente (*smart city*), par exemple. Cela doit notamment permettre de s'assurer que la modélisation de la topologie dynamique via les p-graphes dynamiques conserve autant d'intérêt dans des systèmes plus complexes.

Des applications véhiculaires de nature différente devraient également être utilisées pour tester la méthode de prédiction proposée au cours de la thèse. En effet, des applications de gestion du trafic ou de la pollution, par exemple, peuvent avoir des caractéristiques différentes des applications de sécurité routière. Elles peuvent par exemple impliquer des véhicules plus nombreux et plus distants, utiliser des passerelles vers le réseau internet plus fréquemment ou encore utiliser des données plus anciennes. La prédiction de leur comportement peut s'affranchir des garanties de sécurité routières ce qui autorise plus de liberté dans l'étude de leur algorithme.

Enfin, la modélisation utilisée a démontré qu'elle permettait d'exprimer simplement des conditions réalistes sur la topologie des réseaux véhiculaires. Son utilisation dans le cadre d'études formelles d'algorithmes de coopération véhiculaire (preuves algorithmiques) a le potentiel de simplifier l'expression d'hypothèses sur la dynamique du réseau, tout en utilisant des hypothèses plus réalistes. C'est pourquoi il est envisagé d'utiliser ce modèle dans le cadre de preuves algorithmiques sur des primitives assurant des fonctionnalités simples mais utiles dans les réseaux de véhicules. De telles preuves seraient à même d'augmenter la confiance qu'un acteur de la route peut avoir dans le fonctionnement d'une application de coopération véhiculaire qui les utilise.

Bibliographie

[1] Accidentologie des pietons : Enjeux et recommandations. https ://conseilnational-securiteroutiere.fr/wp-content/uploads/2019/07/ComiteExperts-Accidentologie-pietons.pdf. consulté le 2022-02-08.

[2] Airplug. https://airplug.hds.utc.fr/. consulté le 2021-01-30.

[3] Attention alerte contresens. https ://radio.vinci-autoroutes.com/article/attention-alerte-contresens-5915. consulté le 2022-11-14.

[4] Car 2 Car Communication Consortium. https://www.car-2-car.org. consulté le 2022-11-05.

[5] European telecommunications standards institute. https://www.etsi.org. consulté le 2022-11-05.

[6] European truck platooning challenge 2016. https ://www.government.nl/documents/leaflets/2015/10/06/leaflet-european-truck-platooning-challenge-2016. consulté le 2022-11-05.

[7] Historique de la sécurité routière. https://www.onisr.securite-routiere.gouv.fr/politique-de-securite-routiere/historique-de-la-securite-routiere. consulté le 2022-11-14.

[8] Manuel du conducteur. https://www.jaguar.fr/Images/JJM_11_02_40_901_XF_tcm96-41204_tcm661-534755.pdf. consulté le 2022-12-17.

[9] Network simulator. https://www.nsnam.org/accessed:2021-11-02. consulté le 2021-11-02.

[10] Omnet++. https://omnetpp.org/accessed:2021-11-02. consulté le 2021-11-02.

[11] The one the opportunistic network environment simulator. https://akeranen.github.io/the-one/accessed:2021-11-02. consulté le 2021-11-02.

[12] OpenStreetMap. https://www.openstreetmap.org/. Online ; accessed 7 October 2022.

[13] Passenger transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Passenger_transport_statistics&oldid=274593#Road_passengers. consulté le 2022-12-15.

[14] Simulation of urban mobility. https://www.eclipse.org/sumo/accessed:2021-11-02. consulté le 2021-11-02.

[15] Value of trade in goods by mode of transport. https://ec.europa.eu/eurostat/statistics-explained/images/0/0c/Value_of_extra-EU_trade_in_goods%2C_by_mode_of_transport%2C_2002_and_2021_Dec.png. consulté le 2022-12-15.

[16] Waze. `https://www.waze.com/fr/company`. consulté le 2022-10-20.

[17] Claudia Ackermann, Matthias Beggiato, Sarah Schubert, and Josef F Krems. An experimental study to investigate design and assessment criteria : What is important for communication between pedestrians and automated vehicles ? *Applied ergonomics*, 75 :272–282, 2019.

[18] Aymeric Agon-Rambosson, Jonathan Lejeune, Julien Sopena, and Pierre Sens. Alternating MPR : a balanced broadcast algorithm for MANETs. In *2022 IEEE 21st International Symposium on Network Computing and Applications (NCA)*, volume 21, pages 19–26. IEEE, 2022.

[19] Huda Al Amri, Mehran Abolhasan, and Tadeusz Wysocki. Scalability of MANET routing protocols for heterogeneous and homogenous networks. *Computers & Electrical Engineering*, 36(4) :752–765, 2010.

[20] Karine Altisen, Stéphane Devismes, Anaïs Durand, Colette Johnen, and Franck Petit. Self-stabilizing systems in spite of high dynamics. In *International Conference on Distributed Computing and Networking 2021*, pages 156–165, 2021.

[21] José Javier Anaya, Pierre Merdrignac, Oyunchimeg Shagdar, Fawzi Nashashibi, and José E Naranjo. Vehicle to pedestrian communications for protection of vulnerable road users. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1037–1042. IEEE, 2014.

[22] B. Awerbuch and S. Even. Efficient and reliable broadcast is achievable in an eventually connected network. In *Proceedings of the third annual ACM symposium on Principles of distributed computing*, pages 278–281, 1984.

[23] Lejla Banjanovic-Mehmedovic, Edin Halilovic, Ivan Bosankic, Mehmed Kantardzic, and Suad Kasapovic. Autonomous vehicle-to-vehicle (V2V) decision making in roundabout using game theory. *Int. J. Adv. Comput. Sci. Appl*, 7(8) :292–298, 2016.

[24] Matthieu Barjon, Arnaud Casteigts, Serge Chaumette, Colette Johnen, Yessin M Neggaz, and Antonella Santone. Maintaining a distributed spanning forest in highly dynamic networks. *The Computer Journal*, 62(2) :231–246, 2019.

[25] Sokratis Barmpounakis, George Tsiatsios, Michael Papadakis, Evangelos Mitsianis, Nikolaos Koursioumpas, and Nancy Alonistioti. Collision avoidance in 5G using MEC and NFV : The vulnerable road user safety use case. *Computer Networks*, 172 :107150, 2020.

[26] Guillaume Béduneau and Bertrand Ducourthial. Diffusion fiable dans les réseaux dynamiques. In *21èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (ALGOTEL 2019)*, 2019.

[27] Guillaume Béduneau, Ghada Jaber, and Bertrand Ducourthial. Dynamic p-graphs for predictions in vehicular networks. In *2021 WiMob Short Papers, Posters and Demos Track (WiMob-SPPDT'2021)*, Bologna, Italy, October 2021.

[28] Guillaume Béduneau, Ghada Jaber, and Bertrand Ducourthial. Reliable multidiffusion with limited memory in vehicular networks. In *PerVehicle 2021 : 3rd International Workshop on Pervasive Computing for Vehicular Systems (PerVehicle 2021)*, Kassel, Germany, March 2021.

[29] Guillaume Béduneau, Ghada Jaber, and Bertrand Ducourthial. A method for predicting ITS cooperative applications performances. *Computer Networks*, 216 :109148, 2022.

[30] Guillaume Béduneau, Ghada Jaber, and Bertrand Ducourthial. Validation d'algorithmes coopératifs dans les réseaux de véhicules. In *24èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (ALGOTEL 2022)*, 2022.

[31] Farid Benbadis, Timur Friedman, M Dias De Amorim, and Serge Fdida. GPS-free-free positioning system for wireless sensor networks. In *Second IFIP International Conference on Wireless and Optical Communications Networks, 2005. WOCN 2005.*, pages 541–545. IEEE, 2005.

[32] Vartika Bhandari and Nitin H Vaidya. Reliable broadcast in radio networks with locally bounded failures. *IEEE Transactions on Parallel and Distributed Systems*, 21(6) :801–811, 2009.

[33] Silvia Bonomi, Giovanni Farina, and Sébastien Tixeuil. Reliable broadcast in dynamic networks with locally bounded byzantine failures. In *Stabilization, Safety, and Security of Distributed Systems : 20th International Symposium, SSS 2018, Tokyo, Japan, November 4–7, 2018, Proceedings 20*, pages 170–185. Springer, 2018.

[34] Abdelwahab Boualouache, Sidi-Mohammed Senouci, and Samira Moussaoui. PRIVANET : An efficient pseudonym changing and management framework for vehicular ad-hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(8) :3209–3218, 2020.

[35] Bouziane Brik and Adlen Ksentini. Toward optimal MEC resource dimensioning for a vehicle collision avoidance system : A deep learning approach. *IEEE Network*, 35(3) :74–80, 2021.

[36] Anthony Buisset, Bertrand Ducourthial, Farah El Ali, and Sofiane Khalfallah. Vehicular networks emulation. In *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, pages 1–7. IEEE, 2010.

[37] Ruo Jun Cai, Xue Jun Li, and Peter Han Joo Chong. An evolutionary self-cooperative trust scheme against routing disruptions in MANETs. *IEEE Transactions on Mobile Computing*, 18(1) :42–55, 2018.

[38] Arnaud Casteigts, Paola Flocchini, Bernard Mans, and Nicola Santoro. Measuring temporal lags in delay-tolerant networks. *IEEE Transactions on Computers*, 63(2) :397–410, 2012.

[39] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5) :387–408, 2012.

[40] Guido A Gavilanes Castillo, Edoardo Bonetto, Daniele Brevi, Francesco Scappatura, Anooq Sheikh, and Riccardo Scopigno. Latency assessment of an ITS safety application prototype for protecting crossing pedestrians. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE, 2020.

[41] Victor Chang, L Jegatha Deborah, Balamurugan Balusamy, and PG Shynu. Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks. *Future generation computer systems*, 78 :943–955, 2018.

[42] Myungwhan Choi, Areeya Rubenecia, and Hyo Hyun Choi. Reservation-based cooperative traffic management at an intersection of multi-lane roads. In *2018 International Conference on Information Networking (ICOIN)*, pages 456–460. IEEE, 2018.

[43] Kevin Christensen, Christoph Mertz, Padmanabhan Pillai, Martial Hebert, and Mahadev Satyanarayanan. Towards a distraction-free waze. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pages 15–20, 2019.

[44] Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. *Discrete mathematics*, 86(1-3) :165–177, 1990.

[45] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (OLSR). Technical report, 2003.

[46] IEEE Computer Society LAN/MAN Standards Committee et al. IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11 : Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11ˆ*, 2007.

[47] Thales Teixeira de Almeida, Lucas de Carvalho Gomes, Fernando Molano Ortiz, José Geraldo Ribeiro Júnior, and Luís Henrique MK Costa. Comparative analysis of a vehicular safety application in ns-3 and veins. *IEEE Transactions on Intelligent Transportation Systems*, 23(1) :620–629, 2020.

[48] Debargha Dey, Francesco Walker, Marieke Martens, and Jacques Terken. Gaze patterns in pedestrian interaction with vehicles : Towards effective design of external human-machine interfaces for automated vehicles. In *Proceedings of the 11th international conference on automotive user interfaces and interactive vehicular applications*, pages 369–378, 2019.

[49] G Di Luna, Stefan Dobrev, Paola Flocchini, and Nicola Santoro. Distributed exploration of dynamic rings. *Distributed Computing*, 33(1) :41–67, 2020.

[50] Yoann Dieudonné, Bertrand Ducourthial, and Sidi Mohammed Senouci. COL : A data collection protocol for VANET. In *2012 IEEE Intelligent Vehicles Symposium*, pages 711–716. IEEE, 2012.

[51] EW Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11) :643–644, 1974.

[52] Danny Dolev. Unanimity in an unknown and unreliable environment. In *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, pages 159–168. IEEE, 1981.

[53] Bertrand Ducourthial. Designing applications in dynamic networks : The airplug software distribution. In *SAFECOMP 2013-Workshop ASCoMS (Architecting Safety in Collaborative Mobile Systems) of the 32nd International Conference on Computer Safety, Reliability and Security*, 2013.

[54] Bertrand Ducourthial, Yacine Khaled, and Mohamed Shawky. Conditional transmissions, a strategy for highly dynamic vehicular ad hoc networks. In *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–8. IEEE, 2007.

[55] Bertrand Ducourthial and Sofiane Khalfallah. A platform for road experiments. In *VTC Spring 2009-IEEE 69th Vehicular Technology Conference*, pages 1–5. IEEE, 2009.

[56] Bertrand Ducourthial and Ahmed Mouhamadou Wade. Dynamic p-graphs for capturing the dynamics of distributed systems. *Ad Hoc Networks*, 50 :13–22, 2016.

[57] Farah El Ali and Bertrand Ducourthial. A distributed algorithm for path maintaining in dynamic networks. In *DYNAM 2011 : 1st International Workshop on Dynamicity*, 2011.

[58] Jessica Enright, Kitty Meeks, George B Mertzios, and Viktor Zamaraev. Deleting edges to restrict the size of an epidemic in temporal networks. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[59] ETSI. Intelligent transportation system ; vulnerable road users (VRU) awareness ; part 1 : Use cases definition, 2019.

[60] TCITS ETSI. Intelligent transport systems (ITS) ; vehicular communications ; basic set of applications ; part 2 : Specification of cooperative awareness basic service. *Draft ETSI TS*, 20(2011) :448–51, 2011.

[61] TCITS ETSI. Intelligent transport systems ; vehicular communications ; basic set of applications ; part 3 : Specification of decentralized environmental notification basic service, std. *ETSI EN Std*, 302 :637–3, 2013.

[62] TR ETSI. 103 562 v2. 1.1 (2019-12),". *Intelligent Transport Systems (ITS)*.

[63] TR ETSI. 103 562 v2. 1.1 ; intelligent transport systems (ITS) ; vehicular communications ; basic set of applications ; analysis of the collective perception service (CPS) ; release 2. *Standard. European Telecommunications Standards Institute : Sophia Antipolis, France*, 2019.

[64] Afonso Ferreira. Building a reference combinatorial model for MANETs. *IEEE network*, 18(5) :24–29, 2004.

[65] Alessio Filippi, Kees Moerman, Gerardo Daalderop, Paul D Alexander, Franz Schober, and Werner Pfliegl. Ready to roll : Why 802.11 p beats LTE and 5G for V2X. *NXP Semicond., Eindhoven, The, Netherlands, Tech. Rep*, 1, 2016.

[66] Michael J Fischer, Nancy A Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Computing*, 1(1) :26–39, 1986.

[67] Francesco Flammini, Andrea Gaglione, Dániel Tokody, and Dalibor Dobrilović. Virtualization technology for LoRaWAN roaming simulation in smart cities. In *Machine Intelligence and Data Analytics for Sustainable Future Smart Cities*, pages 251–265. Springer, 2021.

[68] Paola Flocchini, Matthew Kellett, Peter C Mason, and Nicola Santoro. Mapping an unfriendly subway system. In *International Conference on Fun with Algorithms*, pages 190–201. Springer, 2010.

[69] Carlos Flores, Pierre Merdrignac, Raoul de Charette, Francisco Navas, Vicente Milanés, and Fawzi Nashashibi. A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities. *IEEE Transactions on Intelligent Transportation Systems*, 20(5) :1837–1846, 2018.

[70] Sally Floyd and Vern Paxson. Difficulties in simulating the Internet. *IEEE/ACm Transactions on Networking*, 9(4) :392–403, 2001.

[71] Greg N Frederickson. Data structures for on-line updating of minimum spanning trees, with applications. *SIAM Journal on Computing*, 14(4) :781–798, 1985.

[72] Ana González-Plaza, César Briso-Rodríguez, and Rafael Gutiérrez-Cantarero. Network emulator for V2X communication systems. In *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5. IEEE, 2019.

[73] Paulo Gouveia, João Neves, Carlos Segarra, Luca Liechti, Shady Issa, Valerio Schiavoni, and Miguel Matos. Kollaps : decentralized and dynamic topology emulation. In *Proceedings of the Fifteenth European Conference on Computer Systems*, pages 1–16, 2020.

[74] Ronald L Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1) :43–57, 1985.

[75] Peter John Green. Implementation of a real-time Rayleigh, Rician and AWGN multipath channel emulator. In *TENCON 2017-2017 IEEE Region 10 Conference*, pages 35–39. IEEE, 2017.

[76] Mandeep Kaur Gulati, Monika Sachdeva, and Krishan Kumar. Load balanced and link break prediction routing protocol for mobile ad hoc networks. *J. Commun.*, 12(6) :353–363, 2017.

[77] Azra Habibovic, Victor Malmsten Lundgren, Jonas Andersson, Maria Klingegård, Tobias Lagström, Anna Sirkka, Johan Fagerlönn, Claes Edgren, Rikard Fredriksson, Stas Krupenia, et al. Communicating intent of automated vehicles to pedestrians. *Frontiers in psychology*, 9 :1336, 2018.

[78] Frank Harary and Gopal Gupta. Dynamic graph models. *Mathematical and Computer Modelling*, 25(7) :79–87, 1997.

[79] Alex Hinds, Michael Ngulube, Shaoying Zhu, and Hussain Al-Aqrabi. A review of routing protocols for mobile ad-hoc networks (MANET). *International journal of information and education technology*, 3(1) :1, 2013.

[80] David Ilcinkas and Ahmed Mouhamadou Wade. Exploration of the T-interval-connected dynamic graphs : the case of the ring. In *International Colloquium on Structural Information and Communication Complexity*, pages 13–23. Springer, 2013.

[81] Joshua Joy and Mario Gerla. Internet of vehicles and autonomous connected car-privacy and security issues. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2017.

[82] Koji Kamakura and Bertrand Ducourthial. Experimental validation of cooperative approach near road side units. In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1010–1015. IEEE, 2014.

[83] LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3) :382–401, 1982.

[84] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1) :1–29, 2018.

[85] Sungwon Lee and Dongkyun Kim. An energy efficient vehicle to pedestrian communication method for safety applications. *Wireless Personal Communications*, 86(4) :1845–1856, 2016.

[86] Chi-Yu Li, Giovanni Salinas, Po-Hao Huang, Guan-Hua Tu, Guo-Huang Hsu, and Tien-Yuan Hsieh. V2PSense : Enabling cellular-based V2P collision warning service through mobile sensing. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[87] Zongdian Li, Tao Yu, Ryuichi Fukatsu, Gia Khanh Tran, and Kei Sakaguchi. Towards safe automated driving : Design of software-defined dynamic MmWave V2X networks and PoC implementation. *IEEE Open Journal of Vehicular Technology*, 2 :78–93, 2021.

[88] Arthur L Liestman. Fault-tolerant broadcast graphs. *Networks*, 15(2) :159–171, 1985.

[89] Di Liu, Chuanhe Huang, Xi Chen, and Xiaohua Jia. Space-terrestrial integrated mobility management via named data networking. *Tsinghua Science and Technology*, 23(4) :431–439, 2018.

[90] Lei Liu, Chen Chen, Tie Qiu, Mengyuan Zhang, Siyu Li, and Bin Zhou. A data dissemination scheme based on clustering and probabilistic broadcasting in VANETs. *Vehicular Communications*, 13 :78–88, 2018.

[91] Zhiqiang Lv, Jianbo Li, Chuanhao Dong, and Wei Zhao. A deep spatial-temporal network for vehicle trajectory prediction. In *International Conference on Wireless Algorithms, Systems, and Applications*, pages 359–369. Springer, 2020.

[92] Wanjing Ma, Dabin Liao, Yue Liu, and Hong Kam Lo. Optimization of pedestrian phase patterns and signal timings for isolated intersection. *Transportation Research Part C : Emerging Technologies*, 58 :502–514, 2015.

[93] Marco Malinverno, Giuseppe Avino, Claudio Casetti, Carla-Fabiana Chiasserini, Francesco Malandrino, and Salvatore Scarpina. Performance analysis of c-V2I-based automotive collision avoidance. In *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–9. IEEE, 2018.

[94] Pierre Merdrignac, Oyunchimeg Shagdar, and Fawzi Nashashibi. Fusion of perception and V2P communication systems for the safety of vulnerable road users. *IEEE Transactions on Intelligent Transportation Systems*, 18(7) :1740–1751, 2016.

[95] Vicente Milanés, Javier Alonso, Laurent Bouraoui, and Jeroen Ploeg. Cooperative maneuvering in close environments among cybercars and dual-mode cars. *IEEE Transactions on Intelligent Transportation Systems*, 12(1) :15–24, 2011.

[96] Shima Mohseni, Rosilah Hassan, Ahmed Patel, and Rozilawati Razali. Comparative review study of reactive and proactive routing protocols in MANETs. In *4th IEEE International Conference on Digital ecosystems and technologies*, pages 304–309. IEEE, 2010.

[97] Julian Monteiro, Alfredo Goldman, and Afonso Ferreira. Performance evaluation of dynamic networks using an evolving graph combinatorial model. In *2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 173–180. IEEE, 2006.

[98] M. De Amorim N. Belblidia, L. Henrique M. Costa, J. Leguay, and C. Vania. PACS : Chopping and shuffling large contents for faster opportunistic dissemination. In *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS)*, pages 9–16, Bardonecchia, Italy, January 2011. IEEE.

[99] Frederik Naujoks, Heidi Grattenthaler, Alexandra Neukum, Galia Weidl, and Dominik Petrich. Effectiveness of advisory warnings based on cooperative perception. *IET intelligent transport systems*, 9(6) :606–617, 2015.

[100] V. Naumov, R. Baumann, and T. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119, 2006.

[101] Quang-Huy Nguyen, Michel Morold, Klaus David, and Falko Dressler. Adaptive safety context information for vulnerable road users with MEC support. In *2019 15th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 28–35. IEEE, 2019.

[102] Andrzej Pelc. Fault-tolerant broadcasting and gossiping in communication networks. *Networks : An International Journal*, 28(3) :143–156, 1996.

[103] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. RFC3561 : Ad hoc on-demand distance vector (AODV) routing, 2003.

[104] Radia Perlman. An algorithm for distributed computation of a spanningtree in an extended LAN. *ACM SIGCOMM computer communication review*, 15(4) :44–53, 1985.

[105] Jiayu Qi and Tianhan Gao. A privacy-preserving authentication and pseudonym revocation scheme for VANETs. *IEEE Access*, 8 :177693–177707, 2020.

[106] Pooya Rahimian, Elizabeth E O'Neal, Shiwen Zhou, Jodie M Plumert, and Joseph K Kearney. Harnessing vehicle-to-pedestrian (V2P) communication technology : sending traffic warnings to texting pedestrians. *Human factors*, 60(6) :833–843, 2018.

[107] N. Rajput. Measurement of IEEE 802.11 p performance for basic safety messages in vehicular communications. In *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–4. IEEE, 2018.

[108] Ali Tauseef Reza, T Anil Kumar, and T Sivakumar. Position prediction based multicast routing (PPMR) using Kalman filter over VANET. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 198–206. IEEE, 2016.

[109] Dahlia Sam, Cyrilraj Velanganni, and T Esther Evangelin. A vehicle control system using a time synchronized hybrid VANET to reduce road accidents caused by human error. *Vehicular communications*, 6 :17–28, 2016.

[110] Nicola Santoro, Walter Quattrociocchi, Paola Flocchini, Arnaud Casteigts, and Frédéric Amblard. Time-varying graphs and social network analysis : Temporal indicators and metrics. In *Workshop on Social Networks And MultiAgent Systems (SNAMAS)*, pages 32–38, 2011.

[111] Adrian Segall. Distributed network protocols. *IEEE transactions on Information Theory*, 29(1) :23–35, 1983.

[112] M Selvi and B Ramakrishnan. Lion optimization algorithm (LOA)-based reliable emergency message broadcasting system in VANET. *Soft Computing*, 24(14) :10415–10432, 2020.

[113] Parag Sewalkar, Silvia Krug, and Jochen Seitz. Towards 802.11 p-based vehicle-to-pedestrian communication for crash prevention systems. In *2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 404–409. IEEE, 2017.

[114] Parag Sewalkar and Jochen Seitz. Vehicle-to-pedestrian communication for vulnerable road users : Survey, design considerations, and challenges. *Sensors*, 19(2) :358, 2019.

[115] Mao Shan, Karan Narula, Yung Fei Wong, Stewart Worrall, Malik Khan, Paul Alexander, and Eduardo Nebot. Demonstrations of cooperative perception : safety and robustness in connected and automated vehicle operations. *Sensors*, 21(1) :200, 2021.

[116] Roy Sumner, Bruce Eisenhart, John Baker, et al. SAE J2735 standard : applying the systems engineering process. Technical report, United States. Department of Transportation. Intelligent Transportation ..., 2013.

[117] Amin Tahmasbi-Sarvestani, Hossein Nourkhiz Mahjoub, Yaser P Fallah, Ehsan Moradi-Pari, and Oubada Abuchaar. Implementation and evaluation of a cooperative vehicle-to-pedestrian safety application. *IEEE Intelligent Transportation Systems Magazine*, 9(4) :62–75, 2017.

[118] Manabu Tsukada, Masahiro Kitazawa, Takaharu Oi, Hideya Ochiai, and Hiroshi Esaki. Cooperative awareness using roadside unit networks in mixed traffic. In *2019 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE, 2019.

[119] Andrei Vladyko, Abdukodir Khakimov, Ammar Muthanna, Abdelhamied A Ateya, and Andrey Koucheryavy. Distributed edge computing to assist ultra-low-latency VANET applications. *Future Internet*, 11(6) :128, 2019.

[120] Lei-lei Wang, Zhi-gang Chen, and Jia Wu. Vehicle trajectory prediction algorithm in vehicular network. *Wireless Networks*, 25(4) :2143–2156, 2019.

[121] Xinlan Wang, Xiaodong Cai, Qingsong Zhou, and Jialiang Liu. NUIM : An algorithm for maximizing the influence of information diffusion on dynamic social networks. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 818–822. IEEE, 2022.

[122] Thomas Williams, Paul Alves, Gerard Lachapelle, and Chaminda Basnayake. Evaluation of gps-based methods of relative positioning for automotive safety applications. *Transportation research part C : emerging technologies*, 23 :98–108, 2012.

[123] Myounggyu Won, Aawesh Shrestha, Kyung-Joon Park, and Yongsoon Eun. Safercross : Enhancing pedestrian safety using embedded sensors of smartphone. *IEEE Access*, 8 :49657–49670, 2020.

[124] Xinzhou Wu, Radovan Miucic, Sichao Yang, Samir Al-Stouhi, James Misener, Sue Bai, and Wai-hoi Chan. Cars talk to phones : A DSRC based vehicle-pedestrian safety system. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–7. IEEE, 2014.

[125] Stephen Xia, Daniel de Godoy Peixoto, Bashima Islam, Md Tamzeed Islam, Shahriar Nirjon, Peter R Kinget, and Xiaofan Jiang. Improving pedestrian safety in cities using intelligent wearable systems. *IEEE Internet of Things Journal*, 6(5) :7497–7514, 2019.

[126] B Bui Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02) :267–285, 2003.

[127] Z. Yang and W. Wu. The (T, L)-Path model and algorithms for information dissemination in dynamic networks. *Information*, 9(9) :212, 2018.

[128] Roya Bastani Zadeh, Mehdi Ghatee, and Hamid Reza Eftekhari. Three-phases smartphone-based warning system to protect vulnerable road users under fuzzy

conditions. *IEEE Transactions on Intelligent Transportation Systems*, 19(7) :2086–2098, 2017.

[129] Zhensheng Zhang, Shengbo Chen, and Ju Ren. Opportunistic routing in mobile ad hoc delay-tolerant networks (DTNs). In *Advances in Delay-Tolerant Networks (DTNs)*, pages 179–194. Elsevier, 2021.

[130] Yifei Zou, Dongxiao Yu, Jiguo Yu, Yong Zhang, Falko Dressler, and Xiuzhen Cheng. Distributed byzantine-resilient multiple-message dissemination in wireless networks. *IEEE/ACM Transactions on Networking*, 29(4) :1662–1675, 2021.