

Université
de Toulouse

THÈSE

**En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture (RTSA)

Présentée et soutenue par :

Darwin Fabián ASTUDILLO SALINAS

le : vendredi 27 septembre 2013

Titre :

Téléchargement de Contenus dans les réseaux véhiculaires

Download contents in vehicular networks

Ecole doctorale :

Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

M. André-Luc BEYLOT

M. Emmanuel CHAPUT (co-encadrant)

Rapporteurs :

M. Sidi-Mohamed Senouci, Professeur, Université de Bourgogne/ISAT, Nevers.

M. Jérôme Hâri, Maître de Conférences HDR, Eurecom, Sophia-Antipolis

Membre(s) du jury :

M. André-Luc Beylot, Professeur, INPT/ENSEEIH

M. Emmanuel Chaput, Maître de conférences, INPT/ENSEEIH

Mme. Véronique Vèque, Professeur, Université Paris Sud, Orsay

M. Yacine Ghamri-Doudane, Maître de Conférences HDR, ENSIIE, Evry.

M. Cédric Baudoin, Ingénieur, Thalès Alénia Space, Toulouse

Dédicace

À ma bien-aimée Tania, et mes enfants chéries
Katherine et Adrián pour leur accompagnement,
leur temps et leur soutien dans cette aventure
que la vie m'a donné l'opportunité de jouer.

Para mi amada esposa Tania, y mis queridos
hijos Katherine y Adrián por su compañía, su
tiempo, y su apoyo en esta aventura que la vida
me ha dado la oportunidad de llevar a cabo.

Remerciements

Je tiens à remercier en premier lieu les rapporteurs de cette thèse M. Sidi-Mohammed Senouci et M. Jérôme Häerri pour l'attention avec laquelle ils ont lu mon manuscrit et l'intérêt qu'ils ont porté à mes travaux. Je remercie également les membres du jury qui ont accepté d'examiner : Mme. Véronique Vèque, M. Yacine Ghamri-Doudane, et M. Cédric Baudoin. Je souhaite remercier également mon directeur de thèse M. Andre-Luc Beylot et mon codirecteur M. Emmanuel Chaput pour leur patience, le temps qu'ils m'ont consacré et les connaissances qu'ils m'ont permis d'acquérir.

Les travaux présentés dans ce mémoire ont été réalisés au sein de l'équipe Ingénierie Réseaux et Télécommunications (IRT) de l'Institut de recherche en informatique de Toulouse (IRIT). Je veux donc remercier le laboratoire ainsi que l'École Nationale Supérieure d'Électrotechnique, d'Électronique, d'Informatique, d'Hydraulique et des Télécommunications (ENSEEIH) de l'Institut National Polytechnique de Toulouse de m'avoir accueilli et accompagné. Je remercie Thales et, à travers le SENESCYT, le gouvernement équatorien, de m'avoir offert cette opportunité.

Durant cette aventure, j'ai eu la chance d'être entouré d'amis. Je veux remercier de façon spéciale Réjane Dalcé qui m'a accompagné et aidé tout au long de mon séjour en France. Je remercie également mes amis du laboratoire comme de l'extérieur pour leur support et leur présence.

Enfin, mes derniers remerciements, et sans doute les plus importants, vont vers ma femme chérie Tania, mes enfants Katherine et Mateo, mes parents, mes soeurs et frères pour leur soutien. Ce objectif n'aurait pas pu être atteint sans eux. Leur amour et encouragement ont été les facteurs les plus importants de cet accomplissement.

Résumé

L'évolution des systèmes de communications sans fil a permis d'envisager de très nombreuses applications pour les systèmes de transport intelligents (ITS). Elles peuvent ou non utiliser une infrastructure et iront de la sécurité routière aux applications de confort du conducteur ou aux jeux en réseaux. La *mise à jour de cartes* constitue de notre point de vue une application représentative dans la mesure où ce n'est pas une application de sécurité en tant que telle, mais qu'en revanche elle peut contribuer à réduire les embouteillages en améliorant l'efficacité dans la prise de décisions des conducteurs. Elle possède des caractéristiques facilement identifiables : volume élevé de données, faible contrainte de délai, possibilité de mise en œuvre par des communications d'infrastructure à véhicule, entre véhicules, et hybrides. L'objectif est que les contenus soient téléchargés intégralement par tous les véhicules en un temps minimal, en utilisant le moins de ressources possible et au moindre coût.

Les solutions qui sont apparues comme les plus adaptées ont concerné l'utilisation de solutions 802.11p avec ou sans infrastructure. Dans le cas de solutions avec infrastructure, un certain nombre de points d'accès diffuseront des informations avec des zones de couverture le plus souvent disjointes. Vu les tailles de zone retenues et/ou le débit consacré à ce type d'applications, le passage devant un seul point d'accès ne suffira pas à télécharger de telles cartes. Il s'agit alors de définir des stratégies de diffusion d'information. Une première étude a consisté à comparer une stratégie unicast à du broadcast/multicast. Cette dernière se révèle largement meilleure. Une combinaison de ces principes n'améliore pas les performances du système, car le débit consacré à la transmission unicast ne compense pas le débit non utilisé par le broadcast. Le problème provient des doublons reçus par les véhicules en passant auprès de plusieurs points d'accès consécutifs.

Afin d'atténuer le phénomène des doublons, nous avons eu recours au Codage Réseau linéaire pseudo-aléatoire. L'idée est que le point d'accès diffuse des combinaisons linéaires de morceaux de fichiers. Le grand nombre de ces combinaisons linéaires réduit

de façon significative ce phénomène. De façon complémentaire, nous avons étudié l'utilisation de communications ad-hoc pour combler les morceaux de fichier manquants, en particulier dans le cas d'absence d'infrastructure. Nous avons vérifié que l'on pouvait atteindre de bons résultats dans ce contexte en fonction de la diversité des morceaux de fichiers appartenant aux véhicules rencontrés.

Mots-clés : VANET, téléchargement, V2V, I2V, 802.11p, codage réseau, ordonnancement, mise à jour.

Abstract

The evolution of wireless communications systems have enabled to consider many applications for Intelligent Transportation Systems (ITS). They may or may not use the infrastructure. They will consider from the traffic safety applications up to the driver's comfort or network games. The *map updates* are, from our point of view, a representative application but in the other hand it can help to reduce congestion in improving efficiency in decision making. It has well-defined characteristics : high volume of data, low delay constraint, possibility of implementation of infrastructure-to-vehicle communications, between vehicles and hybrids. The objective is that the contents are fully downloaded by all vehicles in minimum time, using fewer resources and lower costs.

The solutions that have emerged as the most suitable concerned the use of the technology 802.11p with or without infrastructure. In the case of solutions with infrastructure, a number of access points broadcast information with coverage areas most often disjointed. Given the size of area used and/or flow devoted to this type of applications, the transition to a single access point is not enough to download these maps. It is then to define strategies of information dissemination. A first study was to compare a unicast strategy face to broadcast/multicast strategy. The latter appears largely improved. A combination of these principles does not improve system performance, because the flow devoted to unicast transmission does not compensate for the flow not used by the broadcast. The problem is duplicate chunks received by vehicles passing from several consecutive access points.

To mitigate the phenomenon of duplication, we used the linear network coding pseudorandom. The idea is that the access point broadcasts linear combinations of chunks of files. The large number of these linear combinations significantly reduces this phenomenon. In a complementary manner, we investigated the use of ad hoc communications to fill the missing chunks of file, particularly in the absence of infrastructure. We verified that we could achieve good results in this context based on the diversity of chunks of

files which are owned by the encountered vehicles.

keywords : VANET, download, V2V, I2V, 802.11p, network coding, scheduler, update.

Table des matières

Dédicace	i
Remerciements	iii
Résumé	v
Abstract	vii
Table des matières	ix
Table des figures	xv
Liste des symboles	xix
Abréviations et Acronymes	xxiii
1 Introduction	1
1.1 Contexte et objectifs	1
1.2 Contributions	3
2 Contexte de la thèse	5
2.1 Définition du problème	5
2.2 Architecture	6
2.3 Techniques de communication	7
2.4 Paramètres	8
2.5 Technologies	10
2.5.1 Satellite	10
2.5.2 Mobiles (3G/4G)	11
2.5.3 WRAN	12

TABLE DES MATIÈRES

2.5.4	WiMax	12
2.5.5	WiFi	13
2.5.6	WAVE	14
2.6	Tableau récapitulatif des technologies	15
2.7	Conclusions	16
3	État de l'art	17
3.1	L'impact de la mobilité dans WAVE	18
3.2	Profiter de la connaissance de la topologie	20
3.3	L'impact de la collaboration dans la performance	22
3.4	La Mise en Cache	23
3.5	Réseaux opportunistes	24
3.5.1	La Dissémination des données	24
3.5.2	DTN	26
3.6	Le protocole pair à pair dans le contexte véhiculaire	27
3.7	Le codage de paquets dans le protocole pair à pair	28
3.8	Tableau récapitulatif	31
3.9	Conclusions	32
4	Communications I2V pour la distribution de Contenus : Contexte autoroutier	35
4.1	Description du problème	35
4.2	Modèle du système	35
4.2.1	Scénario	35
4.2.2	La topologie réseau	36
4.2.3	Le modèle applicatif	36
4.2.3.1	Scénario de référence : Système surchargé	37
4.2.3.2	Scénario limite : Capacité maximale du système	38
4.2.3.3	Scénario sous charge : Système non saturé	38
4.3	Métriques de performances	38
4.4	Évaluation analytique du Goodput du système	39
4.4.1	Modèle de Bianchi	39
4.4.2	Évaluation de la densité en tenant compte de la mobilité	42
4.4.3	Prise en compte de la densité dans le modèle	43
4.5	Paramètres de simulation	44
4.6	Résultats	45

TABLE DES MATIÈRES

4.6.1	Le Goodput du système	45
4.6.2	Taux de délivrance de paquets	46
4.6.3	Délai	47
4.6.4	Équité	48
4.6.5	Fragmentation	48
4.7	Conclusions	50
5	Passage à l'échelle - Comparaison unicast multicast	53
5.1	Description du problème	53
5.2	Modèle du système	53
5.2.1	Modèle de mobilité	53
5.2.2	La topologie réseau	54
5.2.3	Le modèle applicatif	55
5.2.3.1	L'application en unicast	55
5.2.3.2	L'application en multicast	55
5.2.3.3	Combinaison de l'unicast et du multicast	56
5.2.3.4	Le morceau le plus recherché en premier	57
5.3	Métriques de performances	58
5.4	Paramètres de simulation	58
5.5	Modèle analytique de la stratégie d'ordonnancement aléatoire sur un système multicast	59
5.5.1	Analyse du nombre de nouvelles transmissions	59
5.5.2	Analyse du nombre de nouvelles réceptions	60
5.5.3	Analyse complète avec plusieurs RSUs	60
5.6	Résultats de simulation	61
5.6.1	Multicast vs unicast avec une seule RSU	61
5.6.2	Le comportement de « multicast » sur un scénario avec plusieurs RSU	61
5.6.2.1	Stratégie d'ordonnancement Séquentielle	62
5.6.2.2	Stratégie d'ordonnancement Aléatoire	64
5.6.2.3	Utilisation conjointe des schémas multicast et unicast	66
5.6.2.4	Multicast avec la méthode du morceau le plus recherché d'abord	66
5.7	Conclusions	69

TABLE DES MATIÈRES

6	Codage Réseau pour le téléchargement des données par infrastructure	71
6.1	Description du Problème	71
6.1.1	Le problème du multicast : morceaux de fichier doublons	71
6.1.2	Amélioration du taux de réception	72
6.1.3	Les techniques fondées sur le feedback	72
6.1.4	Les techniques fondées sur la diminution de sources de corrélation	72
6.2	Modèle du Système	73
6.2.1	Modèle de Mobilité	73
6.2.2	La topologie réseau	73
6.2.3	Le modèle applicatif	74
6.3	Introduction à PRAVDA	74
6.3.1	Utilisation du codage réseau	74
6.3.2	Codage réseau pseudo aléatoire	76
6.4	Modèle analytique	76
6.4.1	Nombre de morceaux de fichier reçus	77
6.4.2	Nombre de graines différentes	77
6.4.3	Nombre de morceaux de fichier linéairement indépendants	79
6.5	Paramètres de simulation	80
6.6	Résultats de simulation	80
6.7	Conclusions	82
7	Utilisation des communications V2V pour le téléchargement de fichiers	83
7.1	Description du problème	83
7.2	Modèle du Système	84
7.2.1	Modèle de Mobilité	84
7.2.2	La topologie réseau	85
7.3	Description de l'application	86
7.3.1	Sans Network Coding (NC) dans la partie Infrastructure à Véhicule (I2V)	87
7.3.1.1	Utilisation de l'unicast	87
7.3.1.2	Utilisation du multicast	88
7.3.2	Avec NC dans la partie I2V	89
7.4	Les traces I2V	89
7.5	Métriques de performances	93

TABLE DES MATIÈRES

7.6	Paramètres de simulation	93
7.7	Résultats de simulation	94
7.7.1	Double sens	94
7.7.1.1	Les résultats dans le cas où le NC n'est pas utilisé sur la partie I2V	94
7.7.1.2	Les résultats dans le cas où le NC est utilisé sur la partie I2V	96
7.7.1.3	Analyse de doublons	96
7.7.2	Un seul sens	98
7.8	Conclusions	101
	Conclusions	103
	Références	107
	Liste des publications	115

TABLE DES MATIÈRES

Table des figures

2.1	Ensemble de technologies	7
2.2	Composants de la technologie WAVE	14
4.1	Scénario	36
4.2	Couverture d'une RSU	37
4.3	Scénario	38
4.4	Scénario	38
4.5	P en fonction de T_s	41
4.6	κ en fonction du débit (C), la taille du fichier est de $1KB$	41
4.7	Modèle de Bianchi intégrant l'impact de la variation de la densité	44
4.8	Goodput du système normalisé	46
4.9	Taux de délivrance de paquets en fonction de la densité de véhicules	47
4.10	Délai en fonction de la densité de véhicules	48
4.11	L'équité en fonction de la densité de véhicules	49
4.12	Fragmentation en fonction de la densité de véhicules	50
5.1	Scénario	54
5.2	Vecteur de bitmaps et vecteur de fréquences	57
5.3	Vecteur de bitmaps et vecteur de fréquences après envoi du morceau 4	58
5.4	Taux de réception τ en fonction du flux véhiculaire	61
5.5	τ en fonction de M avec la Stratégie Séquentielle	62
5.6	τ en fonction de M avec la Stratégie Séquentielle, τ est dégroupé par RSU	63
5.7	τ en fonction de M avec la Stratégie Aléatoire	64
5.8	τ en fonction de M avec la Stratégie Aléatoire, τ est dégroupé par RSU ([A] Analytique, [S] Simulation)	65
5.9	Taux de réception τ (évalué analytiquement) en fonction du nombre de RSUs, pour différents taux de pertes de paquets	65

TABLE DES FIGURES

5.10	τ en fonction de M avec la stratégie unicast plus multicast en utilisant l'algorithme séquentiel	67
5.11	τ en fonction de M avec la stratégie unicast plus multicast en utilisant l'algorithme aléatoire	67
5.12	τ en fonction de M avec la stratégie du morceau le plus recherché d'abord	68
5.13	τ en fonction de M avec la stratégie du morceau le plus recherché d'abord, τ est dégroupé par RSU ([R] Aléatoire, [P] Le morceau le plus recherché d'abord)	69
6.1	Scénario	73
6.2	Ratio de l'overhead pour les coefficients incorporés	75
6.3	Probabilité d'avoir M graines différentes quand $M + \varepsilon$ morceaux ont été reçus	77
6.4	Nombre de morceaux différents reçus	78
6.5	Probabilité d'être incapable de reconstruire le fichier en fonction du nombre de morceaux reçus correctement	80
6.6	Taux de réception moyen pour les différentes techniques en fonction de la taille du fichier (M), étiquettes [A] Analytique, [C] Codage Réseau, [R] Aléatoire	81
7.1	Le scénario d'intermittence	84
7.2	Le téléchargement d'un fichier	84
7.3	V2V en complément d'I2V	85
7.4	Scénario simplifié avec une seule zone blanche	85
7.5	Densité de probabilité du nombre de morceaux reçus – I2V sans NC.	90
7.6	Distributions normales en fonction de la taille du fichier – I2V sans NC.	91
7.7	Indice de similarité entre OBU voisines les plus proches circulant dans le même sens de circulation	92
7.8	La distribution normale en fonction de la taille du fichier avec NC	92
7.9	Indice de similarité du bitmap entre OBU voisines circulant dans le même sens	93
7.10	τ en fonction de la taille du fichier	94
7.11	Latence en fonction de la taille du fichier	95
7.12	π_1 en fonction de la taille du fichier	95
7.13	Latence en fonction de la taille du fichier – I2V avec ou sans NC	96
7.14	Pourcentage de doublons en fonction de M	97
7.15	τ en fonction de la taille du fichier	98

TABLE DES FIGURES

7.16 Latence en fonction de la taille du fichier	99
7.17 Pourcentage de doublons en fonction de M	100
7.18 π_1 en fonction de la taille du fichier	100

Liste des symboles

Liste des symboles

α	Le facteur de partage du débit entre l'unicast et le multicast, page 57
a^n	Le vecteur bitmap de n-ème OBU, page 57
b	Le nombre de RSUs, page 60
β	Le nombre de bits de la graine, page 76
\mathbf{B}	Le vecteur de bitmaps, page 57
C	Le débit maximal théorique du support, page 37
c	Distance pendant laquelle les OBUs sont couvertes, page 36
C_a	Le débit applicatif moyen, page 37
C_f	Le débit attribué à un flux, page 37
C_{fo}	Le débit C_f en prenant en compte de l'overhead, page 43
C_M	Le débit utilisé par le multicast, page 57
C_U	Le débit utilisé par l'unicast, page 57
d	Le nombre de morceaux engendrés avec une graine différente, page 77
Δ_t	L'intervalle entre bitmaps envoyés par les OBUs, page 88
e	Un morceau codé, page 74
$E[n]$	L'espérance du nombre de véhicules à portée d'une RSU, page 43
$E[P]$	L'espérance de la taille de la charge utile exprimée en unités de temps, page 40
ε	Le nombre de morceaux additionnel qu'il faut pour décoder le fichier, page 76
$E[t]$	L'espérance du temps de couverture, page 42
f	Le vecteur de fréquences, page 57

Liste des symboles

- G Le nombre de morceaux d'un groupe, page 39
- γ Le nombre de bits de chaque coefficient, page 74
- $\Gamma_{M, M_O, M_R, q}$ La probabilité qu'une OBU reçoive q nouveaux morceaux à travers la zone de transmission d'une RSU, page 60
- HG La probabilité d'une nouvelle transmission, page 60
- κ La normalisation de la relation entre la proportion de temps utilisée par la charge utile et le temps total utilisé par le mécanisme d'accès, y compris « l'overhead », page 40
- κ Le rapport entre la charge utile reçue par l'ensemble des OBUs dans une seconde et le débit théorique du support, page 43
- L La longueur de l'autoroute, page 54
- l Le nombre de morceaux linéairement indépendants, page 79
- Λ La fréquence d'envoi de bitmaps, page 93
- λ Le taux d'arrivée des véhicules, page 42
- M La taille d'un fichier en nombre de morceaux, page 39
- m_i Le morceau avec la numérotation i , page 55
- M_O Le nombre actuel de morceaux de fichier reçus par une OBU, page 60
- M_R le nombre maximum de morceaux de fichiers que l'OBU peut recevoir dans une seule RSU, page 59
- mu_i La vitesse d'entrée d'un véhicule, page 42
- N Le nombre moyen de véhicules à portée d'une RSU, page 36
- N_k Le nombre moyen de véhicules par kilomètre et par chaussée, page 36
- n_M Le nombre de véhicules utilisant une transmission en multicast, page 57
- n_U Le nombre de véhicules utilisant une transmission en unicast, page 57
- P La charge utile en unités de temps, page 40
- p La distance de couverture de la portée d'une RSU, page 36
- π_1 La proportion des OBUs qui ont téléchargé le fichier complet, page 58

$P_i(b)$	La probabilité qu'un véhicule reçoive i morceaux différents après avoir traversé b RSU consécutives, page 60
$\pi_{k,q}$	La probabilité qu'une OBU reçoive correctement q nouveaux morceaux, page 60
p_m	La moitié de la portée d'une RSU, page 36
P_s	La probabilité de succès d'une transmission, page 40
P_{tr}	La probabilité qu'il y ait au moins une transmission par intervalle de temps, page 40
r	Le nombre de morceaux reçus correctement, page 77
\bar{r}	Le nombre moyen de morceaux reçus, page 77
ρ_e	La partie du support réellement consacrée, page 37
ρ_s	La portion du support nécessaire à l'application, page 37
R_i	L' i -ème RSU, page 56
σ_χ	L'écart type des morceaux reçus par les OBUs, page 90
σ_{R_i}	La permutation de morceaux pour la RSU R_i , page 56
σ_μ	L'écart type de la vitesse dans une distribution normale, page 73
ς	La taille de la tranche de temps (slot), page 40
τ	Le taux moyen de réception des véhicules, page 57
τ_s	Le seuil à définir pour basculer du multicast vers l'unicast, page 57
Tc	Le temps pris pour une transmission infructueuse, page 40
Θ	Le débit réellement émis sur le support, page 44
t_i	Le temps de traversée d'un véhicule, page 42
Ts	Le temps pris pour une transmission réussie, page 40
$\bar{\mu}$	La vitesse moyenne en un point, page 42
μ_χ	Le nombre moyen des morceaux reçus par les OBUs, page 90
μ_{max}	La vitesse maximale des véhicules, page 42
μ_{min}	La vitesse minimale des véhicules, page 42
$\Phi(l)$	La probabilité que l morceaux soient linéairement indépendants, page 79

Liste des symboles

- φ La probabilité de densité de morceaux reçus, page 90
- χ Le nombre des morceaux reçus par une OBU, page 90

Abréviations et Acronymes

BP Bundle Protocol. [26](#)

CBR Constant bitrate. [37](#), [44](#)

CCDSV Cooperative Content Distribution System for Vehicles. [21](#)

DP Data Pouring. [25](#)

DSRC Dedicated short-range communications. [1](#), [45](#), [59](#)

DSSS Direct-Sequence Spread Spectrum. [40](#)

DTN Réseau Tolérant aux Délais/Perturbations. [18](#), [26](#)

DVB Digital Video Broadcasting. [10](#)

DVB-SH DVB-Satellite services to Handhelds. [10](#), [16](#)

EDGE Enhanced Data Rates for GSM Evolution. [11](#)

EIRP Equivalent Isotropically Radiated Power. [45](#), [59](#)

eMBMS evolved Multimedia Broadcast Multicast Service. [106](#)

ERTICO Organisation pour les ITS en Europe. [1](#)

ETSI European Telecommunications Standards Institute. [1](#), [10](#), [14](#)

GMR GEO-Mobile Radio Interface. [10](#)

GPRS General Packet Radio Service. [10](#)

GSM Global System for Mobile Communications. [10](#)

HSPA High-Speed Packet Access. [11](#)

I2V Infrastructure à Véhicule. [xii](#), [xiii](#), [xvi](#), [14](#), [17](#), [19–21](#), [27–33](#), [35](#), [72](#), [82–87](#), [89–91](#), [93](#), [94](#), [96](#), [97](#), [101](#), [104–106](#)

Abréviations et Acronymes

- IEEE** Institute of Electrical and Electronics Engineers. 1, 12, 14, 35, 103
- ITS** Systèmes de Transport Intelligents. 1
- IVN** Réseau intervéhiculaires. 23
- MANET** Mobile ad hoc network. 5, 17, 20
- MDDV** Mobility-Centric Data Dissemination Algorithm. 24, 25
- MPRP** Le morceau le plus recherché en premier. 72, 88, 90
- NC** Network Coding. xii, xiii, xvi, 26, 28, 29, 86, 87, 89–91, 94, 96, 98, 99, 101, 106
- NHTSA** National Highway Traffic Safety Administration. 1
- OBU** On-Board Unit. 14–16, 18, 20–25, 29, 30, 35–37, 39, 40, 43, 46–48, 50, 54–60, 62–64, 68, 74, 77, 83–92, 94, 96, 97, 99
- OFDM** Orthogonal frequency-division multiplexing. 40
- P2P** Pair à Pair. 8, 16, 18, 28–32, 86–88
- PDR** Packet Delivery Ratio. 39, 45, 46
- PLR** Taux de perte de paquets. 46
- PRAVDA** Pseudo Random Network Coding in Vanet for Data Download. 73, 74, 80–82
- PRNC** Codage Réseau Pseudo-Aléatoire. 74, 76, 82
- PRNG** Générateur de Nombres Pseudo-Aléatoires. 76
- PSDU** Physical layer Service Data Unit. 40, 45, 59
- RSU** Road Side Unit. 14, 17, 19–22, 25, 29–31, 33, 35–37, 40, 41, 44, 45, 47, 48, 50, 51, 53–57, 59–65, 68, 69, 71–74, 76, 77, 81–87, 89–91, 104–106
- SPAWN** Swarming Protocol For Vehicular Ad-Hoc Wireless Networks. 27, 30, 32
- TCP** Transmission Control Protocol. 19
- UDP** User Datagram Protocol. 19, 37, 87
- UMTS** Universal Mobile Telecommunications System. 11
- V2V** Véhicule à Véhicule. 14, 15, 17, 18, 21, 22, 26–33, 74, 82–87, 90, 95, 101, 104–106
- VADD** Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. 20

VANET Vehicular Ad Hoc NETwork. [1](#), [2](#), [6](#), [17](#), [18](#), [20](#), [26](#), [27](#), [30](#), [74](#)

VDTN Réseau Véhiculaire Tolérant aux Délais/Perturbations. [26](#)

WAVE Wireless Access For Vehicular Environnement. [6](#), [10](#), [14–17](#), [19](#), [106](#)

ZIPPER Zero-Infrastructure Peer to Peer System. [27](#)

Abréviations et Acronymes

Chapitre 1

Introduction

1.1 Contexte et objectifs

Malgré l'amélioration de la qualité des véhicules et des infrastructures, les accidents et la mortalité routière restent des sujets de préoccupation importants. Les autorités s'impliquent fortement et en particulier financièrement dans toutes les solutions qui pourront améliorer la sécurité routière. Les progrès des communications sans fil offrent un ensemble de pistes intéressantes pour faciliter la mise en œuvre de nouvelles fonctionnalités liées à ce contexte. Elles s'inscrivent dans un cadre plus large connu sous le nom de « [Systèmes de Transport Intelligents \(ITS\)](#) ».

Ces communications peuvent s'adosser à une infrastructure de télécommunication existante ou bien se dérouler directement entre les véhicules. Outre les applications de sécurité que nous venons de mentionner, elles permettent également de déployer des applications dites de confort qui peuvent être liées au contexte véhiculaire (informations routières, embouteillages, places de parking) ou non (accès internet, jeux en réseau,...).

Plus spécifiquement, les « [Vehicular Ad Hoc Network \(VANET\)](#) » sont apparus pour prendre en charge la partie dite sans infrastructure de ces communications. L'architecture a été normalisée par « [l'ETSI ITS](#) » en Europe et par la famille des normes « [IEEE 1609](#) » aux États-Unis. La Commission Fédérale de Communications des États-Unis a accordé 75 MHz de spectre dans la bande des 5,9 GHz en 1999, pour le « [Dedicated short-range communications \(DSRC\)](#) » qui est consacré aux « [ITS](#) ». En Europe, en 2008, l'[ETSI](#) a alloué 30 MHz de spectre dans la bande des 5,9 GHz pour les [ITS](#). [DSRC](#) a été intégré dans le groupe de travail 802.11 de l'[IEEE](#) et utilise la norme [IEEE 802.11p](#). La « [National Highway Traffic Safety Administration \(NHTSA\)](#) » aux États-Unis, ainsi qu'[ERTICO](#) en Europe, sont impliqués dans les [ITS](#). Leur objectif

1. INTRODUCTION

est d'améliorer la sécurité des passagers et de réduire les accidents mortels grâce aux applications de sécurité. Ces dernières peuvent utiliser les communications avec ou sans infrastructure. Les applications de sécurité ont été largement étudiées dans la littérature et dans des projets nationaux et internationaux.

Il existe par ailleurs des applications dites de confort, visant à améliorer les communications et intégrant les véhicules dans un réseau ubiquitaire et pervasif. Il s'agit d'une thématique importante et nous lui consacrons le présent mémoire. Ces applications vont transformer le quotidien des passagers. Ils pourraient surfer sur internet, regarder leurs mails, télécharger des fichiers, etc. En outre, les applications embarquées peuvent en profiter pour mettre à jour leurs contenus.

Parmi ces applications, on retrouve l'assistance au voyage qui guide les conducteurs pour atteindre leur destination en se servant de cartes. Cependant, celles-ci doivent être mises à jour pour que le chemin calculé par l'application soit efficace. La mise à jour de cartes constitue donc une application représentative dans la mesure où ce n'est pas une application de sécurité en tant que telle, mais qu'en revanche elle peut contribuer à réduire les embouteillages en améliorant l'efficacité dans la prise de décisions. Les éléments d'une carte peuvent avoir un impact local (une ville) ou global (une région, un pays ou même plusieurs pays). Ils peuvent aussi avoir différentes fréquences de mise à jour ; par exemple, l'ajout des rues peut se faire chaque mois. De plus, chaque élément de la carte sert à une quantité importante de véhicules. Dans la littérature, on trouve le terme « popularité » pour désigner le pourcentage de clients qui demandent de télécharger un fichier. Nous supposons ici que le taux de pénétration (ou « popularité ») de l'application est important, et qu'il justifie l'utilisation de techniques dédiées assurant une utilisation rationnelle des ressources.

Cette application nous est apparue comme emblématique, car elle répond à un besoin réel des utilisateurs, et possède des caractéristiques bien définies : volume élevé de données, faible contrainte de délai, possibilité de mise en œuvre par des communications d'infrastructure à véhicule, entre véhicules, et hybrides. Le transfert de fichiers est toujours une application majeure des réseaux de communication.

Un bref panorama nous a permis d'identifier plusieurs technologies pouvant être utilisées pour le téléchargement de contenus : le satellite, le WiFi, les réseaux mobiles (3G/4G), le WiMax, et bien sûr le réseau [VANET](#).

L'objectif est bien évidemment que les contenus soient téléchargés intégralement par tous les véhicules en un temps minimal, en utilisant le moins de ressources possible et au moindre coût. Notre objectif sera donc d'étudier ces différentes configurations et leur combinaison pour mettre en évidence un certain nombre de recommandations pour

atteindre ces objectifs.

1.2 Contributions

La première contribution de la thèse concerne une analyse critique des différentes solutions techniques permettant de mettre en œuvre une telle application. Il s’agissait en particulier d’étudier les divers moyens de communication envisageables : satellite, mobiles, sans fil, mais encore le mode de communication : avec ou sans infrastructure, unicast/broadcast/multicast. Nous avons proposé un certain nombre de métriques permettant d’écarter a priori un certain nombre d’entre elles en raison de leur inadéquation ou de leur coût prohibitif. Les solutions qui sont apparues comme les plus adaptées ont concerné l’utilisation de solutions 802.11p avec ou sans infrastructure.

Dans le cas de solutions avec infrastructure, un certain nombre de points d’accès diffuseront des informations avec des zones de couverture le plus souvent disjointes. Vu les tailles de zone retenues et/ou le débit consacré à ce type d’applications, il apparaît que le passage devant un seul point d’accès ne suffira pas à télécharger de telles cartes (fichier de grande taille). Il s’agit alors de définir des stratégies de diffusion d’information.

Une première étude a consisté à comparer une stratégie de broadcast/multicast à de l’unicast. Il apparaît alors clairement que le broadcast/multicast conduit à des taux de réception de messages bien supérieurs. Une combinaison de ces principes n’améliore pas les performances du système, car le débit consacré à la transmission unicast ne compense pas le débit non utilisé par le broadcast. Le problème provient des doublons reçus par les véhicules en passant auprès de plusieurs points d’accès consécutifs.

Notre contribution suivante a donc consisté à essayer d’éliminer ce phénomène. Pour cela, nous avons eu recours au Codage Réseau linéaire pseudo-aléatoire. L’idée est que le point d’accès diffuse des combinaisons linéaires de morceaux de fichiers. Le grand nombre de ces combinaisons linéaires réduit de façon significative les risques de doublons. Un récepteur peut efficacement recevoir un nombre suffisant de combinaisons qui soient linéairement indépendantes.

De façon complémentaire, nous avons étudié l’utilisation de communications véhicule à véhicule pour combler les morceaux de fichier manquants, en particulier dans le cas d’absence d’infrastructure. Nous avons vérifié que l’on pouvait atteindre de bons résultats dans ce contexte en fonction de la diversité des morceaux de fichiers possédés par les véhicules rencontrés.

1. INTRODUCTION

Chapitre 2

Contexte de la thèse

2.1 Définition du problème

Il existe une grande variété d'applications qui, à court ou moyen terme, vont pouvoir être utilisées dans un environnement véhiculaire. Certaines visent à améliorer la sécurité routière. D'autres cherchent à profiter des différentes technologies pour rendre confortable le séjour dans la voiture, surtout s'il s'agit d'un long voyage.

Dans le contexte de cette étude, comme nous venons de l'expliquer dans l'introduction, nous nous focalisons sur l'aspect confort, en prenant comme exemple le téléchargement de contenus. Plusieurs travaux se sont attaqués à ce sujet, en proposant des adaptations de certains protocoles existants dans les réseaux filaires ou bien encore dans les « [Mobile ad hoc networks \(MANETs\)](#) ». Nous nous intéressons dans cette thèse à l'utilisation de nouvelles techniques et technologies pour améliorer la performance du téléchargement dans les environnements véhiculaires. Nous proposons donc d'approfondir le transfert de fichiers volumineux.

Les travaux que nous allons mener portent sur les applications de téléchargement de contenus ayant un grand nombre de récepteurs ; l'exemple le plus significatif correspond aux opérations de mise à jour, par exemple d'une carte routière, ou d'un logiciel embarqué. Les caractéristiques de ce type d'application sont : les mises à jour assez fréquentes, la variabilité de la taille des informations à télécharger, leur degré d'urgence relativement faible et leur grand nombre d'utilisateurs. Il faut ajouter à cela les contraintes liées à l'environnement véhiculaire : mobilité élevée, perte de signal potentiellement fréquente en fonction de la technologie utilisée qui se traduit par une intermittence dans la connexion en l'absence d'une infrastructure dense. Le fait que les dispositifs à bord ne soient pas activés tout le temps entre aussi en ligne de compte, le véhicule devant

2. CONTEXTE DE LA THÈSE

être démarré.

Les cartes, qui sont le cœur de ce système, sont celles qui nécessitent le plus de mises à jour. Elles contiennent certains *éléments* (rues, aéroports, places, etc.) qui changent plus souvent que d'autres. Ces éléments peuvent avoir leur *fréquence de mise à jour* propre, leur degré d'importance et de pertinence, leur impact (global, régional, local), et leur date limite au-delà de laquelle l'élément concerné n'a plus d'importance.

Dans le contexte véhiculaire, les communications étant sans fil, le principal souci en plus de réduire la charge du serveur de fichiers, sera de limiter le gaspillage de bande passante. Ce gaspillage, dans l'application proposée, peut provenir de la transmission de nombreux flux unicast en réponse aux demandes des clients. Globalement, l'ensemble des flux engendre des répétitions de paquets qui ne sont pas nécessairement envoyés en même temps, car les clients peuvent commencer leur flux à des instants différents. De plus, des retransmissions sont éventuellement nécessaires. La mise en œuvre d'un tel système avec des flux unicast rencontre un problème de passage à l'échelle à cause des nombreux utilisateurs concurrents qui souhaitent faire des mises à jour.

Il convient alors de définir plus précisément les technologies sans fil à utiliser, car la structure et les performances dépendent fortement de ces dernières. Par exemple, nous avons mentionné le fait que certains contenus peuvent avoir un impact global, le recours à une technologie qui ne permette pas une telle couverture peut provoquer une utilisation inefficace des ressources. Plusieurs technologies peuvent être prises en considération comme nous allons le voir maintenant.

2.2 Architecture

La figure 2.1 montre les différentes technologies dont l'application pourrait se servir. Elle indique également leur portée respective. L'architecture est constituée d'un serveur, de véhicules (utilisateurs avec une mobilité élevée), de points d'accès (WiFi, VANET), de satellite, d'antennes (DVB-SH et 3G/4G).

L'ensemble de technologies dont l'application pourrait se servir est composé du satellite, les réseaux mobiles (3G/4G), le WiFi, le WiMax, le WRAN et le [Wireless Access For Vehicular Environnement \(WAVE\)](#)¹.

Les protocoles de communications utilisés sont ceux de la pile TCP/IP. La couche physique et la couche d'accès au support sont définies dans chaque technologie.

1. Ce terme couvre les communications d'infrastructure à véhicule et de véhicule à véhicule. Bien que VANET soit souvent utilisé pour désigner les deux, il ne correspond réellement qu'aux réseaux véhiculaires ad-hoc.

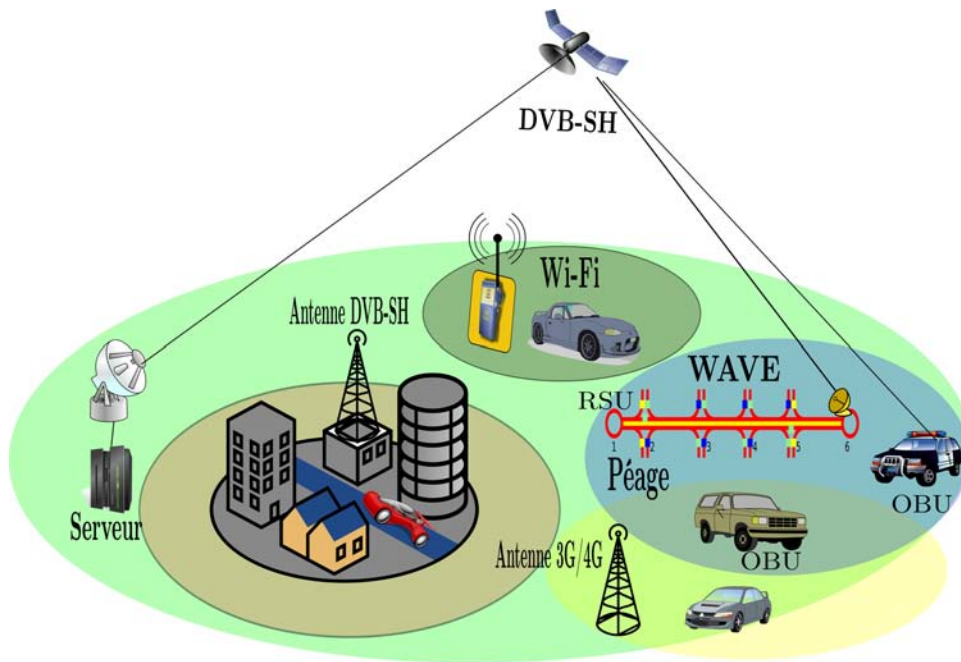


FIGURE 2.1 – Ensemble de technologies

Les technologies peuvent utiliser des communications avec et sans infrastructure (*ad-hoc*). Cette dernière peut s’organiser sans avoir besoin d’une infrastructure définie préalablement. Chacune possède un rôle et comportement différents dans l’architecture. Si une technologie ne propose pas de mode *ad-hoc*, les utilisateurs sont obligés d’utiliser l’infrastructure pour communiquer entre eux. Les technologies utilisant une infrastructure, sauf le WiFi, sont opérées et possèdent des bandes de fréquence réservées.

Il faut considérer aussi que certaines technologies disposent uniquement d’une communication unidirectionnelle et la prise en charge du canal retour est mise en œuvre par une autre technologie.

2.3 Techniques de communication

Les *schémas de délivrance* pouvant être utilisés sont l’unicast et/ou le multicast/-broadcast. L’*unicast* impose la présence d’un flux par utilisateur. Pour des applications de distribution de contenus en masse, ce schéma peut devenir très coûteux. Le *Multicast/Broadcast* permet d’envoyer les paquets à un ensemble d’utilisateurs intéressés en utilisant un seul flux. La possibilité d’utiliser le multicast sur une technologie per-

2. CONTEXTE DE LA THÈSE

met un passage à l'échelle efficace et surtout de profiter de la propriété diffusante du support hertzien. Le broadcast se traduira par une diffusion potentiellement répétée à tous les utilisateurs. La gestion du multicast sera plus fine avec une diffusion vers les utilisateurs concernés qui pourra s'interrompre si plus personne n'est intéressé.

Ces différents schémas présentent donc des avantages et des limites bien connues. Le choix est a priori difficile à faire et constituera une partie de ce travail de thèse.

Deux *types de communications* peuvent être utilisés en fonction de la technologie. Le type *Infrastructure* possède des antennes qui servent de relais entre utilisateurs, ou entre un utilisateur et le réseau d'accès. La communication dans le type *ad-hoc* peut se réaliser de client à client à condition qu'ils soient à portée.

L'application peut utiliser les *paradigmes* « Client - Serveur » et « **Pair à Pair (P2P)** ». Le paradigme *Client - Serveur* est utilisé naturellement dans une configuration avec infrastructure. Le serveur envoie les données vers les clients. Le paradigme **P2P** peut être utilisé sur un réseau de type infrastructure et sur un réseau ad-hoc. Le protocole **P2P** sur un réseau hertzien peut profiter de la topologie physique pour améliorer ses performances.

Nos travaux de thèse vont porter sur ces techniques qui peuvent être de plus complémentaires et croisées avec les technologies sous-jacentes. Le nombre de combinaisons devenant explosif, nous allons tenter d'en éliminer un certain nombre.

2.4 Paramètres

Plusieurs paramètres permettent de déterminer la contribution d'une technologie aux performances et peuvent être regroupés en deux familles. Les paramètres intrinsèques font référence au coût, à la couverture, au débit, et finalement, à la latence qui est fonction des paramètres précédents à l'exception du coût. Les paramètres extrinsèques comprennent entre autres le taux de pénétration dans le marché, la fréquence de mise à jour et l'impact des contenus. Décrivons maintenant ces paramètres.

- Le *coût* de la technologie peut faire pencher la balance pour qu'un client décide de l'utiliser ou non. Nous avons trois types de coût liés au client : la communication, le dispositif, et le service (par exemple un abonnement de téléchargement de cartes). Le coût de la mise en œuvre de la technologie est lié au fournisseur.
- La *zone de couverture* est un paramètre important, car si nous avons une couverture limitée, comme dans le cas d'un point d'accès WiFi (quelques dizaines de mètres), les véhicules qui vont relativement vite vont avoir peu de temps pour interagir avec le point d'accès.

-
- La *latence* est le temps pris par un client pour télécharger un fichier. La satisfaction du client est fondée sur ce paramètre. Plus le téléchargement sera rapide, plus le client sera satisfait. Cependant, il existe un compromis entre la latence et le coût.
 - Le *débit* fait référence à la quantité de données téléchargées par unité de temps. Plus le débit sera grand, plus faible sera la latence.
 - Le *taux de pénétration* est un taux mesurant la couverture du marché par une technologie.
 - L'*impact* fait référence à la zone de couverture effective dont un élément a besoin (global, régional, local). L'impact d'un élément d'une carte d'un point de vue « zone » n'est pas négligeable puisqu'il va entraîner la prise de décision de la technologie à utiliser. Cet impact peut être local ou au contraire couvrir plusieurs pays. Pour le déterminer, il faut considérer les différents abonnements des utilisateurs et le fait que même si l'utilisateur reste la plupart du temps dans une région, il peut se déplacer vers d'autres régions.

Par exemple, l'ajout d'une nouvelle rue se caractérise par un impact local et une faible fréquence de mise à jour. Donc, même si l'impact est local, la fréquence permet de regrouper plusieurs rues ajoutées dans une région particulière (un pays, un département) et les envoyer une fois par mois en utilisant une technologie qui permette de couvrir cette région, par exemple le satellite. De cette façon, l'ensemble des utilisateurs reçoit les mises à jour de leur zone ainsi que les mises à jour de tout le pays. Une autre solution consiste à envoyer, en utilisant par exemple la technologie mobile, les mises à jour relatives à une zone, car les utilisateurs ne voudraient pas payer pour des données qu'ils utiliseraient peu ou pas. Dans le cas du changement de sens d'une rue, deux actions sont à accomplir : la première consiste à mettre à jour à court terme les utilisateurs locaux, et la seconde à mettre à jour le reste des utilisateurs avec une latence moins contraignante.

D'autres paramètres intrinsèques sont les schémas de délivrance possibles et la prise en charge de vitesses de déplacement élevées. Cependant, ils sont proposés par l'ensemble des technologies que nous allons analyser. En outre, nous devons également tenir compte du fait que les bandes de fréquence utilisées par les technologies peuvent être partagées, ou dédiées à une technologie particulière et/ou aux réseaux véhiculaires.

Dans la section architecture 2.2, nous avons mentionné qu'une technologie peut se servir d'une communication ad-hoc. La prise en charge de cette dernière permet l'échange entre véhicules, et cela a un impact plus important lorsque les véhicules sont hors de la zone de portée de l'infrastructure d'une technologie.

2. CONTEXTE DE LA THÈSE

Les vitesses élevées sont le principal inconvénient dans un environnement véhiculaire. Elles réduisent les performances des technologies. Ainsi, lorsque la couverture de ces dernières est limitée, les ruptures de connectivité se multiplient, car les durées pendant lesquelles les équipements sont à portée sont courtes.

2.5 Technologies

Pour la mise en œuvre de cette application dans un environnement véhiculaire, nous pouvons utiliser plusieurs technologies, la technologie sans fil spécifique pour cet environnement repose sur la norme 802.11p. Cette dernière décrit les spécifications des couches physique et MAC pour le « [WAVE](#) ». Chacune d'elles présente des avantages et inconvénients vis-à-vis des exigences de notre application.

2.5.1 Satellite

Comme nous l'avons dit auparavant, la forte *mobilité* est une contrainte non négligeable lors du choix de la technologie. Les satellites peuvent présenter de grands avantages dans le contexte que nous nous proposons d'étudier en raison de leur large *couverture*. En revanche, ils sont connus pour leurs *coûts* potentiellement élevés. Parmi les technologies satellite que nous pouvons utiliser, nous retenons en particulier « [GEO-Mobile Radio Interface \(GMR\)](#) »² et « [DVB-Satellite services to Handhelds \(DVB-SH\)](#) »³.

[GMR](#)[1] propose un service mobile par satellite. Il est considéré comme une extension de [GSM](#), sa version terrestre. La version satellite du [GPRS](#) est appelée GmPRS, elle dispose d'un *débit* maximal de 432 *Kbit/s*. Elle utilise le schéma *unicast* pour le téléchargement de données, ce qui compromet le passage à l'échelle. Par ailleurs, son *coût* est rédhibitoire en termes d'équipements, mais aussi de diffusion. Les territoires couverts par exemple par leurs cousins terrestres sont largement suffisants et à moindre coût. Nous ne retiendrons donc pas [GMR](#) ni GmPRS pour l'application que nous proposons.

[DVB-SH](#)[2] est l'adaptation de la norme [DVB](#) pour les dispositifs portables. Elle est conçue pour la transmission de vidéo, audio et données à des véhicules et des dispositifs portables. Le *débit* minimum est de 833 *Kbit/s* et le maximum de 17,2 *Mbit/s* au niveau de la couche physique. Le schéma utilisé est le *broadcast*. Cette technologie est adaptée

2. Il s'agit d'une norme [ETSI](#) pour la téléphonie par satellite

3. Il s'agit d'une norme de couche physique pour la délivrance de contenus de type multimédia ou données à des terminaux portables tels que les téléphones mobiles ou PDA

aux données qui ont besoin d'une grande *couverture*. Pour l'application proposée, le passage à l'échelle est donc facile. En outre, elle est peu sensible à la mobilité. La *latence*, en utilisant cette technologie, est fonction uniquement du débit et de la taille du fichier.

Avec cette technologie, il faut tenir compte de la disponibilité des utilisateurs au moment de la diffusion. Pour atteindre la totalité des utilisateurs, il faut introduire un système de répétition qui va renvoyer les mêmes contenus. Une étude de ce type a été menée par Soares et al. [3]. Certaines heuristiques peuvent lui être ajoutées : par exemple, la plupart des voitures seront disponibles aux heures de pointe. Cette répétition va introduire une latence supplémentaire.

Il faut mentionner que cette technologie ne permet que des communications unidirectionnelles, pour la voie de retour nous sommes obligés d'utiliser une autre technologie.

En 2013, cette technologie possède malheureusement un faible taux de pénétration dans le marché. En revanche, les grandes marques de mobiles intelligents envisagent de l'intégrer dans les années à venir.

Nous avons étudié l'intérêt d'une telle technique [75], ces travaux n'ont pas été poursuivis en raison du coût important d'une telle technologie qui ne peut s'envisager que dans des contextes de niches (flotte de véhicules, application militaire ou de sécurité civile).

2.5.2 Mobiles (3G/4G)

La technologie 2G/3G offre un débit de téléchargement maximal de 384 Kbit/s (pour de vitesses de 3 km/h à 100 km/h) et de 144 Kbit/s ($< 250\text{ km/h}$) en utilisant **EDGE**. Avec **UMTS** il est de 2 Mbit/s ; en **HSPA** beaucoup plus et finalement la 4G offre un débit de 100 Mbit/s pour les utilisateurs à forte mobilité. Ces débits sont toutefois purement théoriques, car le support est partagé entre les utilisateurs d'une même cellule.

Le coût de téléchargement par mega-octet reste son talon d'Achille. Le *débit* et la grande *couverture* pourraient compenser cet aspect pour certains types de contenus, par exemple, s'il s'agit de contenus prioritaires (latence faible) et qu'il n'y a pas de technologie moins onéreuse disponible pour la remplacer. Ce cas de figure risque en effet de se produire pendant quelques années. De plus, les connexions sont stables grâce à l'utilisation de fréquences réservées pour cette technologie.

2. CONTEXTE DE LA THÈSE

2.5.3 WRAN

Cette architecture est définie par la norme [IEEE 802.22](#)[4]. Cette norme regroupe les spécifications pour la mise en œuvre de réseaux régionaux sans fil. Elle utilise les fréquences VHF et UHF et profite de fréquences qui ne sont pas utilisées par les canaux de télévision (fréquences réservées). Sa *portée* peut aller jusqu'à la centaine de kilomètres. Son débit minimum au niveau de la couche physique atteint $3,74 \text{ Mbit/s}$ et au maximum $24,96 \text{ Mbit/s}$.

WRAN tolère la mobilité des nœuds. Cependant, elle ne prend pas en charge les transferts intercellulaires (version 2010). Grâce à sa large couverture, elle est peu influencée par la mobilité. Comme dans le cas de la technologie satellite, sa *latence* est fonction du débit et de la taille du fichier. Cependant, si les véhicules se trouvent en bordure de la zone de couverture, sa latence va aussi dépendre de la mobilité des véhicules, spécifiquement de la façon dont les véhicules entrent et sortent de la zone de couverture.

Les *coûts* d'implantation de l'infrastructure et des dispositifs restent élevés. En revanche, le coût d'accès reste raisonnable puisqu'il s'agit d'une technologie principalement destinée à desservir les régions rurales et éloignées où l'ADSL arrive peu ou pas. Aujourd'hui sa couverture reste faible puisqu'il s'agit d'une norme relativement nouvelle.

2.5.4 WiMax

WiMax est définie par la norme [IEEE 802.16](#)[5]. Sa couverture peut atteindre 60 km . Son débit minimum est de 110 Mbit/s et le maximum de 365 Mbit/s dans la version 2 de la norme. Elle a été conçue pour fournir le *dernier kilomètre* aux clients fixes dans un premier temps. Dans sa dernière version, la norme prend en charge les utilisateurs mobiles, mais avec un débit maximum de 100 Mbit/s . Elle pourrait être utilisée par les points d'accès et par les véhicules comme un accès de *dernier kilomètre*. La latence se comporte de la même façon que dans les WRAN, mais elle sera plus affectée par la mobilité puisque WiMax possède une couverture moins importante que les WRAN.

Pour le moment le coût d'implantation est élevé autant pour les fournisseurs que pour les clients. Cependant, la stabilité de la connexion est garantie grâce à l'utilisation de fréquences réservées. Son taux de pénétration, bien que plus important que dans le cas des WRAN, reste toujours faible.

2.5.5 WiFi

Cette technologie utilise potentiellement plusieurs normes telles que 802.11 a/b/g/n/ac[6, 7]. Elle peut être utile à des véhicules se trouvant dans une station de service, une aire de services, un restaurant, ou bien encore, utiliser les connexions résidentielles ouvertes (sans mécanisme d'authentification), et les hotspots (avec authentification à partir de paramètres connus). Les utilisateurs se trouvant chez eux ou à leur travail peuvent bien évidemment utiliser leur propre réseau WiFi. La couverture va jusqu'à une centaine de mètres en fonction de la norme et de la configuration utilisée. Le débit peut aller de 1 *Mbit/s* jusqu'à 11 *Mbit/s* pour 802.11b, de 6 *Mbit/s* jusqu'à 54 *Mbit/s* pour 802.11a/g, jusqu'à 300 *Mbit/s* pour 802.11n, et avec 802.11ac jusqu'à 1 *Gbit/s*.

Les véhicules, s'arrêtant quelques minutes dans les aires de services, peuvent profiter de ces réseaux si l'alimentation électrique du véhicule n'est pas coupée. Cela permettra de proposer aux utilisateurs le téléchargement de mises à jour, d'informations liées à l'aire de services, de publicité, etc. Donc, la taille des contenus téléchargés est restreinte par la disponibilité d'un point d'accès à portée du véhicule et par le débit auquel il peut télécharger. Cependant, s'il y a une transmission de données en broadcast en cours, le véhicule pourrait recevoir ces informations, cette transmission pourrait être une mise à jour ou un fichier populaire qui intéresse l'utilisateur.

La latence est fonction du débit, de la taille du fichier et de la mobilité des véhicules. Cette latence pourrait être grande puisque si l'utilisateur n'arrive pas à recevoir tout le fichier pendant qu'il est dans l'aire de services, le véhicule doit attendre que l'utilisateur s'arrête à nouveau pour terminer ou changer de technologie pour continuer à recevoir.

Bien qu'elle ait un taux de pénétration élevé, la plupart des points d'accès sont privés et protégés par un système d'authentification. Cependant, il existe aussi des points d'accès publics. Les fréquences utilisées par cette technologie sont ouvertes, ce qui est un problème vis-à-vis du partage du support, puisqu'il n'est pas utilisé uniquement par les véhicules, mais par un nombre plus important d'utilisateurs.

Toutes les technologies mentionnées auparavant sont opérées, cela veut dire que les utilisateurs ont besoin d'une infrastructure pour pouvoir communiquer. Si ces derniers sont hors de portée de cette infrastructure, ils seront isolés. En revanche, WiFi peut fonctionner en mode ad-hoc, les véhicules peuvent interagir entre eux sans avoir besoin d'un point d'accès. Ils peuvent profiter de cet avantage pour élargir la zone de couverture d'un point d'accès en tirant parti d'une communication à plusieurs sauts et aussi pour échanger des paquets entre utilisateurs.

2. CONTEXTE DE LA THÈSE

2.5.6 WAVE

Avant de commencer à décrire cette technologie, nous allons définir deux acronymes importants à retenir dans l'architecture WAVE. « On-Board Unit (OBU) » et Road Side Unit (RSU). Les deux types de communications pris en charge par cette technologie, comme dans le cas de WiFi. La communication avec infrastructure, dans laquelle l'interaction s'effectue entre une RSU et une OBU, est connue sous le nom d'« I2V ». La communication de type ad-hoc où les communications se déroulent entre OBUs est appelée « Véhicule à Véhicule (V2V) ». La figure 2.2 montre les composants de la technologie WAVE et les deux types de communication possibles.

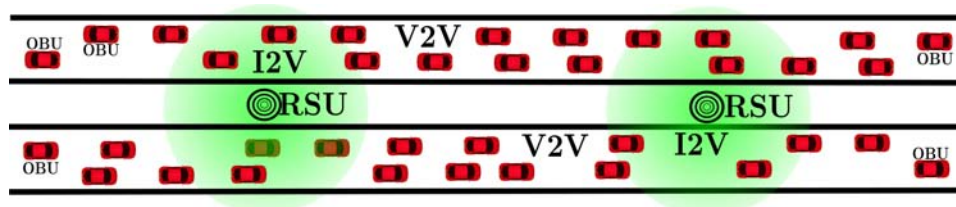


FIGURE 2.2 – Composants de la technologie WAVE

WAVE utilise la norme IEEE 802.11p[8] qui est assez similaire à la norme 802.11a. Son débit minimum est de 3 Mbit/s et le maximum de 27 Mbit/s au niveau de la couche physique. Une RSU dispose d'une portée d'environ 1000 mètres avec le débit minimum et la puissance maximale autorisée. La norme ETSI[9] standardise les allocations de canaux en Europe, tout comme la norme IEEE 1609 le fait aux États-Unis. Dans les deux standards, il existe deux types de canaux : un canal de type contrôle et les autres de type service.

L'intermittence provoquée par le faible déploiement d'infrastructures liées à cette technologie entraîne une latence importante pour les fichiers qui ne peuvent pas être reçus lors du passage auprès d'une seule RSU, ou bien encore lorsque la transmission commence au moment de la sortie du véhicule de l'aire de couverture.

En 2013, le taux de pénétration reste toujours faible et le prix des dispositifs élevé. Cependant, dans les années à venir, comme ce fut le cas avec le WiFi, cette technique pourrait être largement utilisée. Il est important de souligner que la fréquence utilisée est réservée aux réseaux de véhicules ce qui réduit les interférences. Le coût d'utilisation de l'infrastructure pourrait être fonction de l'application. Pour les applications de sécurité routière, le coût serait nul. Pour le téléchargement de mises à jour de cartes, le plus probable est que les fournisseurs de cartes payent l'accès à l'infrastructure et l'utilisateur

l'abonnement. Ces coûts seraient uniquement imputés lors d'une utilisation effective de l'infrastructure. En revanche, l'utilisation de communications V2V pourrait être gratuite.

En profitant du haut débit offert par le 802.11n, la technologie WiFi peut être un complément de la technologie WAVE. Les véhicules possédant les deux interfaces (WiFi et WAVE) peuvent profiter d'une plus grande bande passante et aussi d'un coût nul pour la communication V2V (ad-hoc).

Dans ce document nous utiliserons indifféremment les termes : véhicule, voiture et OBU.

2.6 Tableau récapitulatif des technologies

Le tableau 2.1 montre les paramètres que nous avons jugés les plus importants.

Technologies	Paramètres							
	Coût	Couverture	Débit Minimal	Débit Maximal	Mobilité affecte la latence	Fréquence dédiée à WAVE	Taux de pénétration	Ad-hoc
DVB-SH	C	Pays	833 <i>Kbit/s</i>	17,2 <i>Mbit/s</i>	✗	✗	F	✗
3G/4G	A ⁴	Pays	144 <i>Kbit/s</i>	100 <i>Mbit/s</i>	✗	✗	É	✗
WRAN	C	≈ 100 <i>km</i>	3,74 <i>Mbit/s</i>	24,96 <i>Mbit/s</i>	P	✗	F	✗
WiMax	C	≈ 50 <i>km</i>		100 <i>Mbit/s</i>	P	✗	F	✗
WiFi	A	≈ 100 <i>m</i>	1 <i>Mbit/s</i>	300 <i>Mbit/s</i>	✓	✗	É	✓
WAVE	A	≈ 1 <i>km</i>	3 <i>Mbit/s</i>	27 <i>Mbit/s</i>	✓	✓	F	✓

TABLE 2.1 – [A] Abordable, [C] Cher, [É] Élevé, [P] Peu, [F] Faible

4. Bien que le coût soit abordable, la quantité de données pour télécharger est limitée selon l'opérateur

2.7 Conclusions

Nous pouvons observer que toutes les technologies peuvent contribuer à l'application proposée. Cependant, plusieurs paramètres doivent être pris en compte avant de choisir la solution la plus adéquate.

Bien que le taux de pénétration de la technologie soit un paramètre à ne pas négliger, la technologie [WAVE](#), qui est exclusive pour l'environnement véhiculaire, possède une faible pénétration du marché. Cependant, dans un proche avenir ce taux pourrait croître de façon significative grâce au renouvellement du parc automobile et à l'engagement des constructeurs de fournir de véhicules plus sûrs. Le WiFi et les réseaux mobiles ont déjà un fort taux de pénétration qui peut être utilisé par les applications des environnements véhiculaires.

Finalement, le prix est un autre paramètre à prendre en compte. De plus, celui-ci peut être décisif lors du choix d'une technologie par un utilisateur. Pour le moment, il n'existe pas un modèle économique clair pour la technologie [WAVE](#) et [DVB-SH](#), mais nous sommes convaincus que leur prix sera abordable afin de susciter une réponse du marché (actuellement [DVB-SH](#) reste cher). En ce qui concerne les autres technologies, à l'exception du WiFi, l'utilisateur doit avoir un abonnement dont le prix reste abordable. WiFi apporte une valeur ajoutée aux commerces, son accès est donc gratuit chez la plupart d'eux. En outre, l'application pourrait profiter des hot-spots du fournisseur d'accès auquel l'utilisateur est abonné. En revanche, cela risque de ne fonctionner que dans un environnement urbain.

Si une [OBU](#) avait accès à cet ensemble de technologies, cela permettrait d'avoir une architecture plus adaptée pour cette application où chaque type de contenu pourrait être envoyé au travers de la technologie la plus adaptée.

La conclusion de ce chapitre est que la technologie la plus appropriée à l'environnement véhiculaire et à l'application visée est la solution [WAVE](#) avec et sans infrastructure. Elle est suffisamment générale pour être représentative de la plupart des autres technologies envisagées a priori avec un coût moindre pour l'utilisateur.

Par ailleurs, les schémas de délivrance multicast, broadcast et unicast seront étudiés dans contexte de façon indépendante et, le cas échéant, complémentaire. Pour l'essentiel, le paradigme retenu sera celui du client-serveur en présence d'infrastructure et le [P2P](#) en mode ad-hoc.

Chapitre 3

État de l'art

Dans ce chapitre, nous allons faire un tour d'horizon de différents travaux qui ont été menés dans le contexte des réseaux véhiculaires pour le problème du téléchargement de fichiers.

La dénomination **VANET** qui provient de **MANET** est à la fois méritée en raison des problématiques communes à ces deux types de réseau : support hertzien, nœuds mobiles pouvant servir de relais, mais ne recouvre pas la totalité de la réalité des réseaux véhiculaires. Par exemple, les vitesses élevées des nœuds et la mobilité restreinte à la topologie routière sont très spécifiques dans un environnement **VANET**. Un des problèmes importants est l'intermittence de connectivité, qui est plus prononcée dans les **VANETs** en raison de ces vitesses élevées. En outre, le terme **VANET** correspond uniquement à la communication **V2V**. Le terme **WAVE** comprend en revanche les deux types de communications (**I2V**, **V2V**). Au vu des thématiques que nous avons abordées dans cette thèse, nous devons donc aller au-delà de la terminologie **VANET**. L'environnement véhiculaire ayant des caractéristiques fortes extrêmement spécifiques, nous ne fonderons donc notre structuration de l'état de l'art que sur les propositions faites dans ce contexte.

Notons tout d'abord que pour télécharger un fichier, en particulier de grande taille, nous ne pouvons plus agir comme dans un réseau classique, puisque les communications de bout en bout sont rarement possibles. Les solutions se fondent évidemment sur le profit que l'on pourra tirer de chaque rencontre avec un véhicule ou une **RSU**.

Les premiers travaux dans ce domaine se sont axés sur l'impact de la mobilité dans les communications véhiculaires, sur la délivrance des messages et la faisabilité de communications intermittentes en se servant de communications **V2V** et **I2V**. Nous analyserons des propositions faites en ce sens dans la section 3.1. Cependant, les dé-

3. ÉTAT DE L'ART

placements et la topologie peuvent aussi être une source d'information. Cette dernière peut être utilisée pour améliorer la performance de la délivrance de paquets conjointement avec des heuristiques fondées sur le comportement des utilisateurs comme nous le verrons dans la section 3.2.

La collaboration entre **OBU**s est souvent considérée comme intrinsèque. Les **OBU**s vont collaborer en relayant les messages, le cas échéant en multi-saut. Nous étudierons dans la section 3.3 les techniques fondées sur une telle collaboration. En outre, pour collaborer les **OBU**s doivent mettre à contribution certaines de leurs ressources, à commencer par l'espace mémoire. Il faut donc gérer cet espace pour aider au mieux l'application en utilisant le moins de ressources. Cette procédure est connue sous la dénomination de *mise en cache* et elle joue un rôle important dans la performance du réseau sur certaines applications (cf. section 3.4).

Si nous nous intéressons à l'exploitation de connexions intermittentes dans un contexte d'absence de connectivité de bout en bout, le concept de réseaux opportunistes est particulièrement intéressant. Ils ont été conçus pour permettre aux nœuds de communiquer les uns avec les autres, même si aucune route n'existe jamais entre la source et la destination. Pour le déroulement des communications, les nœuds sont amenés à coopérer afin d'améliorer la qualité de service effective (cf. section 3.5). Dans le contexte de *réseaux opportunistes*, nous pouvons nous intéresser aux solutions qui ont été présentées pour traiter des problèmes similaires dans d'autres domaines. Parmi ces solutions, nous nous focalisons sur celles issues de la conception de réseaux interplanétaires qui ont abouti à ce que nous connaissons aujourd'hui sous le nom de « Réseau Tolérant aux Délais/Perturbations (DTN) ». Nous parlons de ces travaux dans la section 3.5.2.

Nous avons mentionné auparavant les enjeux présents dans les communications véhiculaires. Ils nous donnent un aperçu de la variété des éléments qu'il faut prendre en compte pour utiliser de façon efficace ce réseau. Nous pouvons donc apporter des solutions sur les différentes couches protocolaires. Nous analyserons les travaux qui ont été faits pour proposer un protocole **P2P** plus performant dans des **VANETs** dans la section 3.6 et des propositions variées en utilisant le codage de paquets au niveau applicatif dans la section 3.7.

3.1 L'impact de la mobilité dans WAVE

Un des premiers travaux dans le domaine des communications **V2V** a été réalisé par Chen et al. [10], à l'aide de la technologie WiFi. Ils cherchaient à tester l'hypothèse

selon laquelle les déplacements des véhicules sur une autoroute peuvent contribuer à la délivrance des messages, à condition que ces messages puissent être stockés temporairement en attendant des possibilités de transmission. Ils concluent que le relayage ad-hoc dans les réseaux sans fil offre des opportunités à de nombreuses applications émergentes dans le réseau véhiculaire. Les meilleurs résultats ont été obtenus dans une autoroute à double sens avec plusieurs voies. En outre, avec un faible flux de véhicules les délais peuvent dépasser la seconde ; les applications sensibles au délai ne pourront donc pas s'exécuter. Si on fait référence à la terminologie utilisée par les auteurs, on trouve le terme *stockage* qui correspond à la *mise en cache* de la littérature actuelle.

En se servant d'un réseau WiFi conventionnel dans un environnement véhiculaire, Ott and Kutscher [11] ont mesuré les caractéristiques de transmission pour l'envoi et la réception de gros volumes de données. Ils ont utilisé UDP et TCP dans des véhicules se déplaçant à des vitesses différentes et passant à portée d'une ou plusieurs RSUs. Il s'agit donc d'une communication I2V. Les auteurs ont appelé cette application « Drive-thru Internet ». Elle tire parti de la connectivité intermittente. Il ressort de ce travail que la couverture obtenue à partir d'une seule RSU est beaucoup plus importante que prévue, que l'on peut s'attendre à une faible connectivité aux bordures d'une zone de couverture (dans la phase d'entrée et de sortie), et que TCP se comporte assez bien quand il est utilisé pour des connexions entre nœuds physiquement proches.

Nous pouvons donc conclure de cet état de l'art que la mobilité a un impact fort. Nous ne pouvons pas faire comme si un réseau WAVE était un simple réseau mobile, en particulier parce que la connectivité est trop intermittente, et la durée de contact trop courte.

L'environnement véhiculaire présente une variabilité forte. Cependant, la mise en œuvre d'une application de téléchargement de données est possible. L'application peut profiter des rencontres pour : télécharger depuis un serveur si l'on utilise une infrastructure, et pour échanger de fragments de fichier si des véhicules sont proches. D'autre part, bien que des communications de bout en bout en utilisant TCP soient possibles, elles ne sont pas adaptées à cet environnement. La performance de TCP est dégradée en fonction du nombre de sauts empruntés par la connexion [12, 13].

Pour l'application que nous visons ici (le téléchargement de fichiers), nous utiliserons donc de préférence la couche transport UDP. Elle présente en particulier l'avantage d'un passage à l'échelle plus simple et un impact moins fort sur le serveur lorsque le nombre de clients augmente.

3. ÉTAT DE L'ART

3.2 Profiter de la connaissance de la topologie

La connaissance de la mobilité peut être utilisée pour améliorer la performance des communications. Elle peut jouer un rôle important dans le comportement des applications, car elle est limitée par le tracé des rues, contrairement aux [MANETs](#), où cette restriction est quasiment absente. La densité (ou le facteur de dispersion des véhicules) influence la durée des rencontres entre les véhicules et donc la performance.

Nandan et al. [14] étudient un environnement urbain où les utilisateurs peuvent télécharger un fichier volumineux depuis des [RSUs](#) (communication [I2V](#)). Ils définissent un système permettant d'exploiter les rencontres opportunistes entre les nœuds, augmentant ainsi leur taux de transfert. À cet égard, ils spécifient une technique pour le déploiement des [RSUs](#), fondée sur l'analyse des flux de trafic véhiculaire, qui favorise le téléchargement d'une façon coopérative. De plus, ils proposent et évaluent différents algorithmes pour la sélection des transporteurs et pour l'ordonnancement des morceaux dans le transfert des données, en utilisant la technique « carry and forward ». Finalement, ils analysent le scénario avec une sélection probabiliste de porteurs et un ordonnancement redondant. Ce dernier multiplie par deux dans le pire cas le taux moyen de téléchargement et divise par dix le taux de morceaux non délivrés par rapport à la technique « carry and forward » aveugle.

Pour résoudre le problème de la délivrance de données à plusieurs sauts dans les [VANETs](#), Zhao and Cao [15] proposent « [Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks \(VADD\)](#) ». Ils adoptent l'idée de « carry and forward », où un véhicule en déplacement transporte le paquet jusqu'à ce qu'il rencontre un nouveau véhicule se déplaçant dans son voisinage auquel transmettre la donnée. La différence avec d'autres solutions du même type (carry and forward), repose sur l'utilisation de la prédiction de la mobilité de véhicules, limitée par les schémas de déplacement du trafic routier et le tracé des rues.

Chen and Chan [16] présentent MobTorrent, un système axé sur l'utilisateur et conçu pour les véhicules qui ont un accès intermittent aux [RSUs](#) ; les communications sont néanmoins supposées se dérouler à haut débit. Quand une [OBU](#) veut télécharger des données, à défaut d'attendre un contact avec une [RSU](#), elle informe une ou plusieurs [RSUs](#) pour qu'elles chargent à l'avance des contenus. L'algorithme d'ordonnancement de MobTorrent réplique ensuite les données récupérées à l'avance par les [OBUs](#) assistants, de telle sorte que la quantité totale de données transférées et le taux de transfert moyen pour les [OBUs](#) soient améliorées par rapport à des algorithmes concurrents. Par conséquent, au lieu de limiter le transfert de données à haut débit pendant des périodes

de contact d'une assez courte durée entre les **RSUs** et les **OBUs**, les transferts à haut débit entre les **OBUs** sont exploités de façon opportuniste.

Malandrino et al. [17] étudient un système de téléchargement de contenus qui prend en compte les communications **I2V** et **V2V**, dans un scénario urbain. Pour maximiser la performance du système, ils le formulent comme un problème de flux maximum qui tient compte de plusieurs aspects, tels que la contention du canal et la manière de transférer les données. Ainsi, ils ont identifié les facteurs ayant l'impact le plus important sur la performance et ont élaboré des recommandations pour la conception du réseau véhiculaire et de l'infrastructure routière qui le prend en charge. Les auteurs supposent une connaissance parfaite de la circulation automobile et définissent une solution centralisée optimale. Ce travail a été amélioré dans [18] où ils identifient les limites de la performance du téléchargement avec une modélisation du processus de téléchargement sous forme d'un problème d'optimisation. Ils maximisent le débit global du système et supposent une connaissance a priori des trajectoires de véhicules et un ordonnancement parfait de la transmission des données.

Zhang and Yeo [19] font une introduction au « **Cooperative Content Distribution System for Vehicles (CCDSV)** ». Cette étude s'appuie sur le contexte d'un réseau avec infrastructure. **CCDSV** résout plusieurs problèmes comme le manque de fiabilité de la prédiction de la mobilité, la limitation des ressources dans les **RSUs** et la distribution des contenus de façon partagée. Le système organise les **RSUs** de façon coopérative dans une structure appelée « carte de contacts », qui est fondée sur les motifs de contacts des véhicules observés par les **RSUs**.

Trullols-Cruces et al. [20] considèrent un scénario routier complexe (par exemple, non linéaire) où les utilisateurs sont intéressés par le téléchargement de fichiers volumineux à partir d'une **RSU**. Ils étudient la possibilité d'exploiter des rencontres opportunistes des nœuds mobiles afin d'augmenter le taux de transfert. À cette fin, ils élaborent des solutions pour la sélection des porteurs et des morceaux de données de la **RSU**. Ils les évaluent en utilisant des tracés de routes du monde réel. Différentes stratégies de déploiement des **RSUs** sont testées en s'appuyant sur des cartes des contacts obtenues à partir des historiques des contacts entre flux de véhicules. La carte de contacts est utilisée pour estimer une probabilité de rencontre et choisir le meilleur transporteur possible des paquets à transmettre. Les auteurs montrent que les transferts, en se servant de « carry and forward », peuvent accroître le taux de téléchargement dans un environnement urbain/suburbain. Ce mécanisme est utilisé pour télécharger des fichiers de grande taille. [21, 22] observent des résultats similaires.

De notre point de vue, on peut tirer plusieurs enseignements de ces différents tra-

3. ÉTAT DE L'ART

voux. Tout d'abord, les informations de localisation sont extrêmement importantes pour envoyer des fichiers vers un véhicule particulier, mais ce n'est pas complètement l'objectif que nous visons dans cette thèse. Elles peuvent en revanche aider dans les communications V2V. La prédiction de mobilité est intéressante, car elle peut aider à conserver des contacts longtemps pour échanger des données. Nous pensons qu'il est presque aussi intéressant de communiquer avec plusieurs OBUs consécutivement pour améliorer les chances d'obtenir les données manquantes.

Les historiques de contact enfin ont une importance pour les cas de figure où l'on aurait des données à échanger régulièrement (tous les jours par exemple) ce n'est pas exactement le contexte applicatif que nous avons retenu.

3.3 L'impact de la collaboration dans la performance

En général, dans les travaux sur les réseaux sans fil, une hypothèse consiste à supposer que tous les véhicules vont collaborer pour faire suivre (relayer) les paquets. Cependant, cela n'est pas toujours le cas. Yuen et al. [23] abordent le problème de la non-coopération entre les nœuds dans le cadre de la distribution de contenus dans les réseaux avec des RSUs mobiles. Ils supposent que tous les nœuds ont un intérêt commun dans les fichiers mis en cache dans les RSUs fixes. En plus de télécharger des fichiers à partir des RSUs fixes, les nœuds agissent en tant que RSUs mobiles¹ et échangent ces fichiers quand les nœuds sont à proximité. Les auteurs proposent l'établissement d'un « contrat social », dans lequel l'échange ne se produit que lorsque chaque nœud peut tirer un avantage de l'échange. Ceci permet une efficacité du système plus élevée par rapport au téléchargement en n'utilisant que les RSUs fixes sans exiger une véritable coopération entre les nœuds. On pourrait déduire que sans *contrat social*, il est fort probable qu'il n'y ait pas de coopération. Cependant, il est utile à chacun de coopérer, car si un nœud ne veut pas le faire, il sera ignoré par les autres et sera finalement le plus affecté par la situation.

Un autre problème est celui de l'hétérogénéité. Chaque véhicule peut mettre à disposition ses différentes ressources (mémoire, processeur, etc). L'hétérogénéité peut être due à la mise à jour constante de la technologie, le développement différent de chaque marque, et la configuration choisie sur un véhicule. Le contrat social est une solution efficace mais qui ne prend pas en compte cette hétérogénéité.

La performance de l'application visée sera meilleure quand les véhicules qui collaborent sont nombreux et que les ressources utilisées sont suffisantes et équitablement

1. Nous utiliserons le terme OBU relayeur au lieu de RSU mobile

réparties. Pour avoir une bonne équité, le mécanisme « Tit for tat » utilisé par BitTorrent, peut être envisagé. Cependant, il pourrait être adapté en prenant en compte les caractéristiques du support sans fil.

Dans chaque technique impliquant une forme de collaboration entre les véhicules, y compris celles que nous proposons, la notion de « contrat social » peut donc être introduite par exemple pour favoriser son déploiement.

3.4 La Mise en Cache

Fiore et al. [24, 25] abordent le problème de la *mise en cache* des informations et la délivrance dans les « Réseau intervéhiculaires (IVN) ». Ils proposent une application, appelée Infoshare, fondée sur le paradigme de « pull » et l'acheminement au niveau applicatif. Cette application exploite la propriété de diffusion du support sans fil pour obtenir un maximum de requêtes d'information propagées entre les OBUs, tandis qu'une politique intelligente de *mise en cache* limite les coûts supplémentaires résultant de demandes inutiles ou en double. Avec cette application, plusieurs paramètres ont été étudiés afin d'évaluer leur impact sur les performances du système. Les auteurs ont conclu, en se servant de simulations, que le taux d'arrivée des véhicules se répercute de manière non négligeable sur les performances du système ; les changements de topologie trop rapides conduisent à une connectivité instable et les changements de topologie trop lents induisent des échanges d'information réduits, en raison de groupes isolés de véhicules. En outre, les résultats de simulation montrent que le rapport entre le taux de demande d'informations et le taux d'effacement du cache joue un rôle fondamental pour encourager l'échange d'informations efficace entre les OBUs. Ils ont également développé un modèle analytique et avec lui, ont étudié les relations entre les paramètres du système et la durée de vie de l'information au long de la route.

Dans [26], les mêmes chercheurs proposent une stratégie de *mise en cache* distribuée où chaque nœud décide s'il faut mettre en cache ou non le contenu, et pour combien de temps ; la stratégie est appelée « Hamlet ». La décision est prise en fonction de la perception d'un nœud de ses proches. Le résultat est la création d'une diversité de contenus dans le groupe de nœuds. De cette façon, un contenu est plus susceptible d'être retrouvé à proximité par un utilisateur qui le demande. La diversité obtenue grâce à cette stratégie, aide à avoir un accès efficace aux contenus. De bonnes performances sont obtenues pour des informations ayant un faible niveau de popularité, ainsi que pour les contenus très populaires. En outre, « Hamlet » s'est montré en mesure de maintenir la cohérence entre les copies de la même information en cache au niveau des différents

3. ÉTAT DE L'ART

nœuds du réseau.

Dans l'application qui nous intéresse, chaque véhicule a besoin de conserver les données reçues jusqu'à la construction du fichier recherché. La question de la conservation en cache se pose donc à une autre échelle : doit-on conserver un fichier dont on n'a plus besoin ? La réponse à cette question aura des conséquences sur les performances globales, mais elle dépend largement de paramètres applicatifs.

3.5 Réseaux opportunistes

Les réseaux opportunistes sont un type de réseau où la communication est remise en question par des contacts sporadiques et intermittents ainsi que par des déconnexions et des reconnexions fréquentes, et où l'hypothèse de l'existence d'un chemin de bout en bout entre la source et la destination est abandonnée.

Dans les réseaux opportunistes chaque nœud agit selon le paradigme « store-carry-and-forward ». Cela suppose que les utilisateurs stockent les données qu'ils produisent ou reçoivent, les transportent jusqu'à ce qu'ils trouvent un transporteur approprié (ou la destination), puis les transmettent. Ainsi, les contacts entre les nœuds sont considérés comme une opportunité pour rapprocher les données de la destination.

3.5.1 La Dissémination des données

La dissémination des données représente un sujet important dans les réseaux opportunistes. Elle est un prolongement naturel du « ré-envoi », habituellement fondé sur un modèle de publication/souscription.

L'application proposée détient des informations qui intéressent la plupart des **OBU**s. Par conséquent, un schéma de broadcast semble plus approprié que de l'unicast. Le principal avantage d'un système broadcast est l'inondation, les véhicules ne sont pas obligés de connaître l'adresse de destination ni un itinéraire vers une destination spécifique. Un paquet d'information se propage alors à travers le réseau vers tous les véhicules qui ont besoin de cette information. Mais, dans un réseau dense, les collisions de paquets posent un grave problème puisqu'il existe un grand nombre de véhicules dans la même zone pouvant retransmettre le paquet en même temps. Ce phénomène est appelé *tempête de diffusion*. Les protocoles de dissémination cherchent à pallier ce problème.

Wu et al. [27] proposent « [Mobility-Centric Data Dissemination Algorithm \(MDDV\)](#) », une approche centrée sur la mobilité pour la dissémination de données dans les réseaux véhiculaires. Elle vise à agir de façon efficace et fiable avec un réseau partitionné et

caractérisé par une mobilité élevée. **MDDV** est conçu pour exploiter la mobilité des véhicules pour la dissémination des données, et combine le ré-envoi opportuniste, le transfert fondé sur la trajectoire et le transfert géographique. Les auteurs ont développé une approche fondée sur une mobilité générique pour la conception d’algorithmes centrés sur les réseaux véhiculaires. Les véhicules effectuent des opérations locales en utilisant des informations relatives à cette zone, l’objectif est d’atteindre collectivement un comportement global envisagé. La fiabilité de la délivrance de messages est améliorée en permettant à plusieurs véhicules de propager activement le même message. Ces derniers sont transmis le long d’une trajectoire prédéfinie géographiquement. Comme aucune connectivité de bout en bout n’est supposée, les véhicules intermédiaires doivent mettre en cache et transmettre des messages de façon opportuniste. À la manière d’un algorithme opportuniste, **MDDV** répond aux questions : qui peut transmettre, quand transmettre, et quand stocker/effacer des messages.

En se fondant sur la structure du réseau et la quantité de données engendrées dans celui-ci, la diffusion conduit facilement à une congestion sévère et réduit de manière significative le taux de délivrance des données. Pour résoudre ces problèmes, Zhao et al. [28] proposent le paradigme de données « **Data Pouring (DP)** » avec *mise en cache*. **DP** exploite la mobilité partiellement prévisible des véhicules, et contrainte par la configuration de la route. Ce paradigme diffuse les informations à partir de la **RSU**, qui se trouve sur la route principale, vers les routes secondaires. Pour cela, les véhicules de la route principale sont utilisés comme *mémoire virtuelle* du serveur. Les auteurs supposent que la densité véhiculaire sur la route est assez grande pour maintenir la connectivité du réseau. Ce travail ne prend pas en compte les fichiers de grande taille. Pour ces derniers, les véhicules peuvent ne pas finir de les télécharger. En effet, le temps pour télécharger un fichier est plus grand que le temps qu’un véhicule reste dans la couverture d’une **RSU**.

Skordylis and Trigoni [29] se concentrent sur le développement de techniques « carry and forward », en tentant de délivrer des données des **OBUs** aux **RSUs** dans un milieu urbain. Ils proposent deux schémas pour la ré-émission de paquets pour les réseaux véhiculaires urbains, D-Greedy et D-MinCost, qui visent à acheminer les paquets vers les **RSUs**. Ces algorithmes s’appuient sur la base de connaissances locale ainsi que sur des statistiques globales du trafic, pour prendre des décisions stratégiques de ré-envoi en minimisant le coût de communication tout en respectant un délai maximal. Ces algorithmes surpassent leurs concurrents en termes de coût de communication, tout en obtenant un taux de délivrance similaire à l’algorithme épidémique[30, 31] pour une large gamme de densités de véhicules et de délais maximaux.

3. ÉTAT DE L'ART

Ye et al. [32, 33] analysent le problème de la dissémination de données dans un environnement V2V, en utilisant le Network Coding (NC), ils estiment le gain obtenu par rapport au schéma de base telle qu'une transmission aléatoire.

C'est majoritairement dans cette famille de techniques que nous pouvons classer notre approche, même si nous ne nous interdisons pas de profiter de résultats offerts dans d'autres familles. Nous proposons des mécanismes dédiés aux spécificités de notre application.

3.5.2 DTN

Le DTN est une architecture de bout en bout qui s'efforce de répondre à des problèmes posés aux réseaux hétérogènes tels que : des intermittences de connectivité, de longs délais et, potentiellement, de forts taux d'erreur. Le routage dans de tels environnements est difficile, car les pairs ont peu d'informations sur l'état du réseau partitionné. Les opportunités de transferts entre les pairs sont souvent rares et d'une durée très limitée. Cette approche, publiée dans le RFC4838 et le RFC5050 utilise une abstraction commune appelée « Bundle Protocol (BP) » pour ces applications. Les principales fonctionnalités de ce BP comprennent : la retransmission « custody-based », la capacité de faire face à une connectivité intermittente, la possibilité de profiter d'une connectivité connue *a priori* (planifiée), qui peut être prédite et opportuniste (en plus de la connectivité ininterrompue), et la définition d'un nouvel adressage fondé sur les identificateurs d'extrémité. L'unité de données du protocole s'appelle « bundle ». Les « Réseaux Véhiculaires Tolérants aux Délais/Perturbations (VDTNs) » sont l'évolution permettant la prise en compte des caractéristiques d'un réseau véhiculaire.

Différents travaux portant sur les DTN ont été proposés dans un environnement véhiculaire. Burgess et al. [34] proposent MaxProp, un protocole de routage des messages DTN. Il est fondé sur un algorithme d'ordonnancement pour les paquets transmis à d'autres pairs et pour les paquets à effacer. Les décisions dans les ordonnanceurs sont prises en fonction de la probabilité d'avoir un chemin vers ses pairs. Cette probabilité se fonde sur des données historiques et des mécanismes complémentaires. Parmi ces mécanismes se trouvent les accusés de réception, les « head-start » pour les nouveaux paquets, et les listes d'intermédiaires précédents. D'autres auteurs ont proposé des adaptations de protocoles existants aux VANETs : FTP@VDTN[35] et WWW@VDTN[36].

Notre approche présente certaines similitudes avec les DTNs. Nous cherchons en effet à transmettre des données au travers d'un réseau caractérisé par une grande variabilité de la topologie et une connectivité très sporadique. Nous nous intéressons en

revanche à des applications de type 1 vers n , si bien que nous évitons certaines difficultés telles que l'adressage ou le routage. D'autres enjeux apparaissent en revanche ; tels que la fiabilisation, auxquels nous proposons des réponses spécifiques.

3.6 Le protocole pair à pair dans le contexte véhiculaire

Das et al. [37] proposent une stratégie de coopération pour la dissémination et le partage de contenus. L'objectif visé est la transmission unidirectionnelle sur une autoroute, et cette étude repose sur l'hypothèse que tous les véhicules sur la route sont des téléchargeurs actifs d'un même contenu. Par ailleurs, la solution réduit les transmissions redondantes en s'appuyant sur la propriété de diffusion du support sans fil. Ce protocole propose de découper un fichier en plusieurs morceaux et de les échanger entre les véhicules à la BitTorrent. Le protocole décrit précédemment est appelé **SPAWN** et traite plusieurs problèmes : la découverte de pairs, la sélection de contenus et la découverte de contenus. Selon les résultats rapportés, ce protocole améliore la performance du réseau par rapport à BitTorrent. Lee et al. [38] et Lee and Yap [39] présentent la mise en œuvre du protocole, appelé CarTorrent. Décrivons succinctement les trois principaux axes traités :

- La *découverte de pairs* utilise une méthode centralisée ainsi qu'une approche distribuée. La technique distribuée exploite les propriétés de diffusion du canal radio qui permet aux nœuds d'accéder aux contenus disponibles chez ses voisins.
- La *sélection de contenus* est déterminée par une stratégie de sélection de pièces, qui est fondée sur le principe du plus rare en premier (utilisé par BitTorrent) et de la proximité des nœuds (prise en compte de la topologie physique). En utilisant cette dernière astuce, le protocole est plus performant que BitTorrent.
- La *découverte de contenus* est mise en œuvre pour communiquer la disponibilité de contenus d'un pair vers ses voisins. Ce mécanisme est connu sous le nom de *mécanisme de bavardage*. Cependant, ce dernier induit une surcharge de paquets. Le problème de **SPAWN** est justement ce mécanisme de bavardage car il peut provoquer une *tempête de diffusion*, qui réduit de façon considérable la performance du réseau. Ce travail considère l'utilisation de communications **I2V** pour télécharger les contenus.

Abuelela and Olariu [40] proposent l'application « **Zero-Infrastructure Peer to Peer System (ZIPPER)** », une solution adaptée aux communications **V2V**. L'objectif de ce système consiste à assurer le partage de données entre les nœuds mobiles du **VANET**. En ce qui concerne la découverte de pairs, ils se fondent sur les informations du canal

3. ÉTAT DE L'ART

de contrôle. Pour la découverte de contenus, l'application envoie une requête avec les morceaux dont il a besoin à ses voisins. Si les voisins à un saut ne possèdent pas les contenus demandés, ils relayent à leurs voisins et ainsi de suite. La découverte est réalisée sans utiliser le mécanisme de bavardage. Elle est déclenchée quand un véhicule veut télécharger un fichier. Finalement, pour la sélection de contenus, l'application utilise des messages de type requête et accusé de réception.

Le téléchargement d'un fichier se déroule en utilisant des informations de contrôle. Ces informations permettent au pair *téléchargeur* de faire connaître au pair *serveur* les morceaux dont il a besoin. S'il existe plusieurs *serveurs*, le demandeur doit connaître et contrôler ce qu'il demande à chacun.

Les études mentionnées précédemment ont analysé le téléchargement de fichiers en utilisant le P2P. Ils se sont concentrés surtout sur les problèmes posés en V2V, laissant de côté ceux de I2V. Le premier travail fait juste mention à l'I2V de façon superficielle. Dans les deux cas, un véhicule demande de télécharger des données, et établit un flux avec les véhicules les plus proches qui ont le fichier. Dans notre application, nous nous sommes focalisés sur un seul fichier et avons analysé les différents mécanismes pour réduire la bande passante utilisée et le temps de téléchargement. Ces techniques P2P pourraient parfaitement compléter notre étude dans le cas où il y aurait plusieurs fichiers. Il faudrait en particulier étudier de façon plus approfondie la signalisation engendrée. Dans notre cas, elle sera en effet largement réduite comme nous le verrons par la suite.

3.7 Le codage de paquets dans le protocole pair à pair

Un des problèmes dans les réseaux P2P est la quantité importante de données de contrôle qu'il faut échanger entre les différents pairs. Le pair téléchargeur établit une communication avec chacun des pairs serveurs, et envoie des requêtes contenant les identifiants des morceaux manquants. Ce problème devient plus important lorsque le support utilisé est partagé.

Le NC a été créé dans l'objectif d'améliorer le débit, l'efficacité et le passage à l'échelle d'un réseau filaire [41]. Il a été choisi pour améliorer les protocoles P2P, spécifiquement pour que les pairs serveurs n'aient pas besoin de connaître les paquets spécifiques manquants d'un pair téléchargeur, en réduisant de cette façon les paquets de contrôle [42, 43]. Il a aussi été utilisé pour améliorer le débit et la robustesse des réseaux sans fil, tant au niveau de la couche MAC que de la couche applicative [44, 45].

Nous présentons ici de manière succincte le fonctionnement du Network Coding

(NC)[46]. Pour coder un fichier, il faut d'abord le diviser en M morceaux. Pour générer un *morceau codé*, il faut réaliser une combinaison linéaire de tous les *morceaux non codés* en utilisant des coefficients aléatoires. Le nombre de coefficients nécessaires est égal à M . Pour qu'un véhicule puisse décoder les morceaux reçus, il doit connaître les coefficients avec lesquels les *morceaux* ont été codés. Donc, le serveur ou l'OBU qui possède le fichier doit envoyer les *morceaux codés* et leurs coefficients. Les coefficients vont former un système linéaire. Pour décoder les morceaux, les équations doivent être différentes, mais aussi linéairement indépendantes.

Donc, bien que le NC résolve les problèmes de la sélection de contenus et de la réception de paquets redondants, il montre potentiellement des faiblesses en ce qui concerne l'*overhead* et la complexité algorithmique.

VANETCODE [47] réalise la distribution de contenus de façon coopérative en utilisant le NC. Il s'appuie sur la propriété de diffusion du support pour accélérer la dissémination de blocs codés sur le voisinage à un saut. Les véhicules, en utilisant le protocole P2P, vont d'abord télécharger, s'ils sont dans des aires de couvertures des RSUs. Ce téléchargement va se faire en unicast : un *morceau codé* différent est envoyé à chaque véhicule. Après avoir quitté l'aire de couverture, les véhicules vont s'échanger les morceaux qu'ils détiennent. Il existe une probabilité élevée que les véhicules aient reçu des *morceaux codés* différents. Il en découle que lorsque chaque nœud réencode ses morceaux et les transmet à ses voisins, ces derniers vont recevoir des morceaux avec des coefficients linéairement indépendants avec une probabilité élevée.

Pour contrebalancer le problème de l'*overhead*, VANETCODE divise le fichier en deux niveaux. De cette façon, le nombre de coefficients nécessaires est réduit de moitié. Les auteurs mentionnent le compromis qui existe entre la taille du morceau et le nombre de morceaux. En utilisant une taille de morceau plus importante, on augmente la probabilité qu'un morceau ne soit pas reçu par le récepteur à cause de la brièveté du contact. Cependant, si un grand nombre de morceaux est présent, le problème se pose du côté de la complexité algorithmique, donc du temps nécessaire pour pouvoir décoder le fichier.

Ce travail part de l'hypothèse que les véhicules voisins vont télécharger le même fichier en unicast. Ils ont négligé la propriété de diffusion du support sans fil pour les communications I2V, mais ils l'ont utilisée en V2V. Les auteurs mentionnent qu'à un instant donné tous les véhicules sont à portée de la RSU. Dans leur scénario, si les véhicules avaient utilisé le multicast, ils auraient pu télécharger tout le fichier sans avoir besoin d'utiliser le V2V. Évidemment, en utilisant l'hypothèse de départ, ils ont une probabilité négligeable que les véhicules aient les mêmes morceaux du fichier. Cepen-

3. ÉTAT DE L'ART

dant, s'ils avaient considéré la propriété de diffusion dès le départ, ils auraient eu un problème de corrélation entre les morceaux que chaque OBU possède.

CodeTorrent [48] utilise la dissémination « épidémique » pour la distribution de contenus. Un nœud crée une description d'un fichier et la diffuse à ses voisins à un saut. Les nœuds apprennent l'existence du fichier à partir de la description transmise par leurs voisins, et diffusent un message de requête du fichier. Le nœud qui possède le fichier émet des paquets codés pour répondre aux requêtes. Les auteurs ont montré que cette méthode induit un temps plus court de téléchargement de fichiers que les protocoles existants dans les VANET à cette époque (par exemple, CarTorrent [39]). Le but est de distribuer des données volumineuses, du type divertissement (par exemple de la musique et des films) à tous les véhicules qui les demandent. La dissémination de ces données ne nécessite pas de contrainte stricte de délai. Globalement, CodeTorrent et VANETCODE sont similaires. Dans les deux cas, les communications I2V et V2V sont utilisées. À la différence de VANETCODE, CodeTorrent utilise le mécanisme à deux niveaux pour réduire une partie de l'overhead. Nous pouvons le considérer comme une évolution de SPAWN.

Les auteurs analysent CodeTorrent avec une taille de fichier réduite vis-à-vis du débit utilisé. Il existe une forte probabilité que, même pour la vitesse la plus élevée utilisée dans l'étude, le fichier soit téléchargé par un véhicule quand il traverse la portée de la RSU. Dans ce travail, l'analyse de performance en fonction de la taille des fichiers et la corrélation existante des morceaux, quand plusieurs véhicules sont dans la zone de couverture, sont absents.

Le problème de la distribution des données à partir d'une RSU à des véhicules sur une autoroute est examiné par Johnson et al. [49]. Ils mettent l'accent sur un scénario simplifié où les nœuds mobiles adoptent une stratégie de coopération ALOHA MAC pour distribuer les contenus fournis par une seule RSU. L'architecture proposée est conçue pour un réseau dense, cependant il échoue avec une mobilité de véhicules réaliste.

Jusque-là, les travaux précédents ont proposé une méthode pour résoudre le problème de la sélection de contenus. Cependant, un nouveau problème a surgi : l'overhead produit par les coefficients nécessaires pour le codage/décodage de morceaux.

Afanasyev et al. [50] proposent de diviser le fichier en générations (morceaux de grande taille). Ces générations vont être divisées en morceaux, et ce sont ces morceaux qui vont être codés. Cependant, ceci nous ramène au problème d'échange de morceaux en utilisant un mécanisme classique P2P. Cela signifie qu'il faut envoyer dans un bitmap les générations de chaque véhicule. Les auteurs de cette étude évaluent les différentes

stratégies de sélection des générations (aléatoire, séquentielle, minimum local, maximum local, minimum du voisinage, maximum du voisinage, minimum global, et maximum global). Plusieurs stratégies locales peuvent être utilisées dans les nœuds pour offrir un effet global approprié. Par ailleurs, ils ont évalué deux stratégies spécifiques (minimum et maximum du voisinage) en utilisant une heuristique. Cette dernière élimine tous les nœuds dont le progrès à chaque génération est plus petit que ce progrès local. La stratégie du minimum du voisinage avec l’heuristique obtient une meilleure performance que les autres stratégies.

Sardari et al. [21] abordent le problème de la distribution d’une grande quantité de données dans un réseau peu dense de véhicules depuis une **RSU**, en utilisant les Codes Fontaines et les communications coopératives en **V2V**. Ils ont étudié le problème de la dissémination fiable à partir de plusieurs sources, où chaque nœud dans le réseau partage une quantité limitée de ses ressources pour coopérer avec les autres. Ils utilisent une communication unicast, la communication entre deux pairs du réseau **P2P** dans un environnement véhiculaire se faisait avec de nœuds relayeurs ayant un tampon fini. Pour éviter de surcharger le canal avec des balises trop nombreuses, un véhicule ajuste son intervalle de balise en fonction de sa vitesse de déplacement. Ils ont appelé cette application Infocast.

L’application emblématique, que nous étudierons, renferme des contraintes spécifiques. Nous pouvons utiliser le codage pour en améliorer la performance. Globalement, les travaux présentés se focalisent sur l’environnement **V2V**. Nous pensons que les communications **I2V** doivent également être prises en compte et que l’utilisation d’une technique homogène entre l’**I2V** et le **V2V** permettra de simplifier l’application et d’en accroître les performances.

3.8 Tableau récapitulatif

Pour classifier les travaux qui parlent de la dissémination de contenus dans un environnement véhiculaire, plusieurs paramètres sont considérés : le type de communication **V2V** ou/et **I2V**, le schéma de délivrance (point à point (unicast) ou point à multipoint (multicast/broadcast)), le paradigme utilisé par l’application (**P2P**, de dissémination), et la mobilité. Ce dernier paramètre fait référence aux travaux qui profitent de la mobilité pour améliorer la performance. Nous fondons notre étude sur l’application de mise à jour de cartes. Donc, nous ne récapitulons, dans le tableau 3.1, que les travaux qui se sont intéressés au niveau de couche applicative. Différents travaux proposent un nombre varié de solutions pour enrichir la performance d’un réseau. Ces travaux se fondent sur

3. ÉTAT DE L'ART

les différentes couches protocolaires.

Technologie	Paramètres						
	V2V	I2V	Unicast	Multicast	P2P	Dissémination	Mobilité
SPAWN[37]	✓	✓	✓	✗	✓	✗	✗
SPAWN+[14]	✓	✓	✓	✗	✓	✗	✓
ZIPPER[40]	✓	✗	✓	✗	✓	✗	✗
Johnson et al. [49]	✓	✓	✗	✓	✓	✓	✗
VANETCODE[47]	✓	✓	✗	✓	✓	✗	✗
CodeTorrent[48]	✓	✓	✓	✗	✓	✗	✗
CodeTorrent+[50]	✓	✓	✓	✗	✓	✗	✗
MDDV[27]	✓	✗	✗	✓	✓	✓	✓
Data Pouring [28]	✓	✓	✗	✓	✗	✓	✓
Infoshare [24]	✓	✓	✗	✓	✗	✓	✗
Infocast [21]	✓	✓	✓	✗	✓	✓	✗
VADD [15]	✓	✓	✓	✗	✗	✓	✓
MobTorrent [16]	✓	✓	✓	✗	✓	✓	✓
CCDSV [19]	✓	✓	✓	✓	✗	✓	✓

TABLE 3.1 – Tableau comparatif

Tous les travaux présentés, dans cet état de l'art, traitent le type de communication **V2V**. Deux travaux n'analysent pas du tout la communication **I2V**. Notons encore que la plupart de ceux qui s'y intéressent, le font de façon superficielle. L'unicast est utilisé particulièrement quand il s'agit d'un système **P2P** dans lequel chaque client peut décider du fichier à télécharger et du moment où le faire. Le multicast est utilisé avec des systèmes qui cherchent à disséminer les données à plusieurs véhicules sans que ceux-ci aient besoin de demander le fichier. Il est certain que connaître la mobilité aide à améliorer la performance dans les communications, mais il faut souligner que les utilisateurs peuvent être réticents à fournir des informations de localisation.

3.9 Conclusions

Comme nous avons pu le constater dans cet état de l'art, un grand nombre de solutions ont été proposées dans la littérature pour la diffusion de contenus dans les

réseaux véhiculaires utilisant plusieurs angles d'attaque.

On s'aperçoit néanmoins que ce qui a été assez largement étudié c'est la desserte finale des véhicules avec en particulier le **V2V**. Au vu de ce tour d'horizon, il nous a semblé tout d'abord judicieux d'étudier plus profondément l'utilisation des communications **I2V** dans un contexte de **RSUs** assez éparses et donc des communications depuis un serveur, avant que le **V2V** puisse prendre le relais. On va s'y attaquer, en commençant par un premier schéma de délivrance qui est le plus simple, l'unicast. Nous comparons ensuite avec du multicast puis avec une combinaison unicast/multicast. Finalement, nous reviendrons sur la complémentarité entre **I2V** et **V2V**.

3. ÉTAT DE L'ART

Chapitre 4

Communications I2V pour la distribution de Contenus : Contexte autoroutier

4.1 Description du problème

L'analyse conduite dans le chapitre précédent plaide pour l'étude d'un premier scénario dans lequel la distribution de fichiers volumineux s'effectue au travers d'une infrastructure I2V. La communication est fondée sur le standard IEEE 802.11p. L'utilisation de ce dernier se traduira par un moindre coût pour les utilisateurs par rapport à une solution 3G/4G. Le schéma de délivrance utilisé dans ce chapitre est l'unicast.

Nous allons donc nous intéresser, dans un premier temps, aux performances de l'envoi de données à travers une RSU vers des OBUs. Tout d'abord, nous décrirons le modèle retenu (cf. section 4.2) puis nous définirons des métriques plus précises que celles introduites dans le chapitre 2. Finalement, nous analyserons les premiers résultats obtenus en comparant un modèle analytique et des simulations.

4.2 Modèle du système

4.2.1 Scénario

Il existe différents scénarios envisageables, mais nous allons nous concentrer sur le cas des autoroutes. Rappelons brièvement leurs caractéristiques principales :

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

- Les vitesses élevées¹.
- L'absence d'intersections, et la présence d'échangeurs.
- La présence de deux chaussées à sens unique, chaque chaussée étant composée d'une ou plusieurs voies de circulation séparées par un terre-plein central.

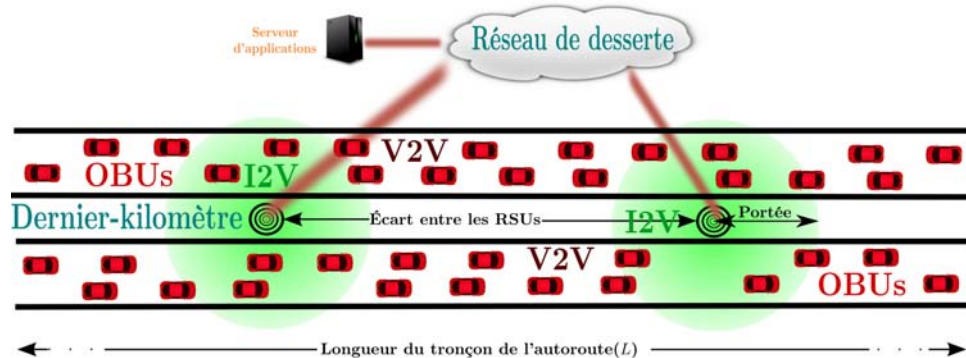


FIGURE 4.1 – Scénario

Le scénario utilisé est composé d'une autoroute avec deux chaussées (une chaussée pour chaque sens), chaque chaussée comportant deux voies (cf. figure 4.1).

4.2.2 La topologie réseau

La topologie est composée par le serveur d'applications, le réseau de desserte et finalement le réseau d'accès constitué de la RSU et les OBU. Dans le serveur, une application fournit les contenus aux OBU (cf. figure 4.1).

4.2.3 Le modèle applicatif

Dans cette étude, nous n'analyserons pas les problèmes liés aux erreurs de transmission de paquets à la périphérie de la zone de couverture. Pour minimiser l'impact de ce phénomène, nous ne considérerons que la moitié de la portée (p_m), donc $p_m = p/2$ où p est la distance de couverture de la portée. Cela signifie que l'application considère uniquement les véhicules étant à portée (p_m). Nous pouvons appliquer cette restriction grâce à l'absence des contraintes temporelles fortes, et l'impact sur la quantité de données téléchargée par une OBU est négligeable. Donc, le nombre moyen de véhicules à portée d'une RSU est $N = N_k \cdot p \cdot 2$ (cf. figure 4.2) où N_k est la densité par kilomètre par chaussée ($veh/km/chaussée$), définie comme paramètre d'entrée.

1. En France cette vitesse est limitée à $130km/h$.

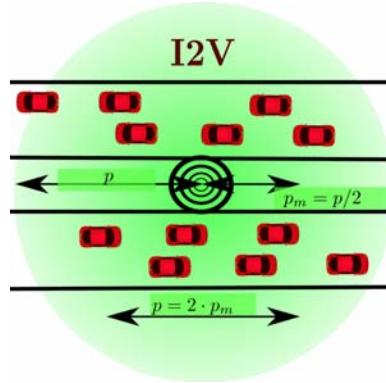


FIGURE 4.2 – Couverture d’une RSU

L’application a été mise en œuvre dans le serveur d’applications. Elle utilise le protocole de transport [UDP](#). Nous avons décidé d’utiliser le mode « [Constant bitrate \(CBR\)](#) » pour l’envoi de données.

Nous désignerons par (C_f) le débit attribué à un flux. Le débit applicatif moyen nécessaire vaut alors $C_a = C_f \cdot N$. Nous notons C le débit maximal théorique du support, $\rho_s = \frac{C_a}{C}$ la portion du support nécessaire à l’application et ρ_e la partie qui y est réellement consacrée tenant compte des erreurs, des pertes...

Dans cette partie de l’étude, la signalisation applicative (requêtes, réponses éventuelles) n’est pas modélisée. Nous nous concentrons sur l’analyse de l’envoi unidirectionnel de données.

4.2.3.1 Scénario de référence : Système surchargé

Dans le premier scénario, le paramètre ρ_e est fixé à 1. Cela signifie que le débit applicatif moyen nécessaire utilise complètement le débit du support. Ceci va provoquer une surcharge dans le système qui résultera en un goulot d’étranglement et en conséquence des pertes. Ces dernières vont se manifester dans la [RSU](#), comme représenté par la figure 4.3.

Ce scénario est utilisé pour atteindre deux objectifs. Le premier consiste à déterminer la *charge du système en sortie* (ρ_s) de ce scénario et de l’utiliser comme *paramètre d’entrée* (ρ_e) dans le deuxième scénario. En effet, dans la mesure où le trafic est essentiellement descendant depuis la [RSU](#) vers les [OBUs](#), les collisions sont absentes et donc pas d’effondrement des performances à forte charge comme nous pouvons le constater dans certains types de réseau sans fil. Cela permettra aussi d’évaluer les différentes

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

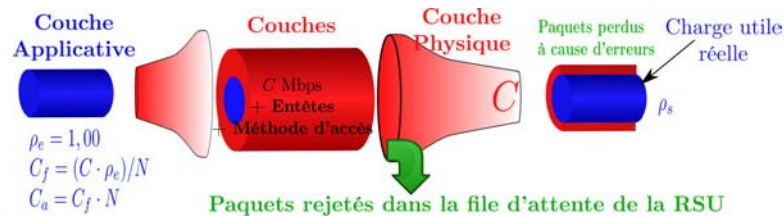


FIGURE 4.3 – Scénario

métriques et de connaître le comportement du système quand il est surchargé.

4.2.3.2 Scénario limite : Capacité maximale du système

Ce scénario utilise comme paramètre la valeur de la métrique ρ_s obtenue dans le scénario surchargé. Donc, $\rho_e = \rho_s$.

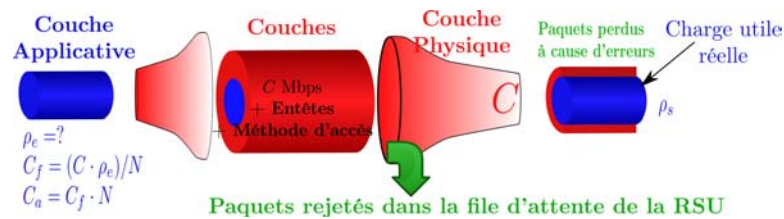


FIGURE 4.4 – Scénario

Nous nous attendons à ne voir subsister que des pertes dues aux erreurs de transmission.

4.2.3.3 Scénario sous charge : Système non saturé

Ce scénario prend une valeur arbitraire pour le paramètre d'entrée ρ_e . Cependant, celle-ci doit être sensiblement inférieure à la charge du scénario précédent.

4.3 Métriques de performances

Afin d'étudier les performances de la première configuration, nous avons retenu les critères de performances suivantes :

- Le « Goodput » : Ce critère permet de connaître la charge utile (payload) délivrée à une certaine destination par unité de temps (payload data rate), au niveau de la couche applicative. Il ne tient donc pas compte de paquets retransmis

au niveau de la couche MAC ni de « l'overhead ». Le « Goodput du Système » est l'agrégation des Goodputs de toutes les destinations. C'est donc ρ_s .

- Le « Taux de délivrance de paquets ou **Packet Delivery Ratio (PDR)** » que nous notons τ : Il permet de déterminer la quantité de données reçue par rapport à la quantité envoyée.
- Le délai : Il spécifie le temps de traitement au niveau de la couche applicative d'un morceau² entre la source et la destination (du serveur jusqu'aux OBUs).
- Le niveau d'équité : Il est représenté par un indice compris entre 0 et 1, 1 correspond au niveau le plus équitable. Nous avons utilisé l'indice de Jain [51]. Dans l'équation (4.1), n représente le nombre de véhicules, et x_i représente le nombre de morceaux reçus par le i -ème véhicule.

$$\mathcal{J}(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2} \quad (4.1)$$

- La fragmentation : Il s'agit d'un indice compris entre 0 et 1, fondé sur la théorie de la percolation [52, 53]. Elle caractérise le niveau de fragmentation d'un fichier. Comme pour l'équité

$$C \equiv \frac{\sum_{j=1}^g G_j(G_j - 1)}{M(M - 1)} \quad (4.2)$$

Ici g est le nombre de groupes (ensemble de morceaux contigus), M est la taille du fichier en nombre de morceaux et G_j est le nombre de morceaux du groupe j .

L'évaluation a été menée par des simulations et par un modèle analytique. Nous commencerons par détailler l'environnement de simulation.

4.4 Évaluation analytique du Goodput du système

Dans la première partie, nous allons résumer le modèle de Bianchi et l'ajuster à nos besoins. Dans la deuxième, nous raffinerons en prenant en compte la mobilité.

4.4.1 Modèle de Bianchi

La charge normalisée obtenue analytiquement montre la relation entre le débit utile et les ressources nécessaires pour la transmettre (en prenant en compte les en-têtes des différentes couches protocolaires et les mécanismes d'accès).

Plusieurs modèles analytiques de 802.11 DCF ont été publiés [54–57]. De nombreux articles sont des extensions du célèbre modèle de Bianchi [58], et essayent de fournir une

2. Nous appellerons morceau les PDU de niveau applicatif.

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

meilleure représentation de la réalité. En effet, l'analyse de Bianchi ne tient compte ni des imperfections du canal, ni de la mobilité des nœuds, ni de la variabilité du nombre de nœuds sur la route (les nœuds entrent et sortent du système). En utilisant des chaînes de Markov, Bianchi modélise le comportement quand le support est saturé. Nous avons utilisé ce modèle avec l'ensemble des paramètres de la norme 802.11p, et calculé la charge ρ_s . La procédure indiquée dans la section 17.3.5.3 du document [59] a été utilisée pour intégrer l'OFDM au lieu du DSSS et pour calculer les temps en fonction de la taille des PSDUs.

Le modèle de Bianchi normalise deux valeurs par unité de temps, ce sont les probabilités de succès de transmission (P_s) et d'au moins une transmission par intervalle de temps (P_{tr}). La première valeur est exprimée en fonction de l'espérance de la taille de la charge utile ($E[P]$) en unités de temps, et des deux probabilités mentionnées, donc $P_s \cdot P_{tr} \cdot E[P]$. La deuxième valeur est la somme de trois éléments : le premier élément est la probabilité d'une tranche de temps vide ($(1 - P_{tr}) \cdot \varsigma$) où ς est la taille de la tranche de temps, le second élément est le temps mis par une transmission réussie ($P_s \cdot P_{tr} \cdot T_s$) et la valeur finale est le temps mis par une transmission infructueuse ($(1 - P_s) \cdot P_{tr} \cdot T_c$). Une transmission réussie avec RTS/CTS comprend le RTS, le CTS, les données et la PPDU ACK. Une transmission infructueuse utilise uniquement la PPDU RTS (le modèle ne prend en compte que les erreurs dues à des collisions). La normalisation donne une relation entre la proportion de temps utilisée par la charge utile et le temps total utilisé par le mécanisme d'accès, y compris « l'overhead ». L'équation (4.3) résume ce paragraphe.

$$\kappa = \frac{P_s \cdot P_{tr} \cdot E[P]}{(1 - P_{tr}) \cdot \varsigma + P_{tr} \cdot P_s \cdot T_s + P_{tr} \cdot (1 - P_s) \cdot T_c} \quad (4.3)$$

L'analyse de Bianchi suppose que tous les clients (RSUs, OBUs) envoient et reçoivent des données (il y a toujours un paquet à transmettre dans la file d'attente de chaque client). Cependant, avec l'application proposée ci-dessus, le seul client à saturer le médium est la RSU. Il n'existe donc pas de collisions. En simplifiant, nous avons la probabilité de succès $P_s = 1$, et la probabilité d'avoir au moins une transmission dans l'intervalle de temps $P_{tr} = 1$. Nous pouvons également simplifier $E[P]$ puisque la charge utile possède une taille constante. Finalement, un rapport entre le temps nécessaire pour transmettre la charge utile P et le temps pris pour transmettre cette charge utile avec l'overhead T_s est obtenu (cf. figure 4.5). La transmission complète du message a une durée T_s . D'où $\kappa = P/T_s$.

Ainsi, la figure 4.6 montre κ pour chaque débit admis par le support en utilisant le



FIGURE 4.5 – P en fonction de T_s

802.11p. La valeur de κ pour un débit de 3 Mbit/s est d'environ $0,775$.

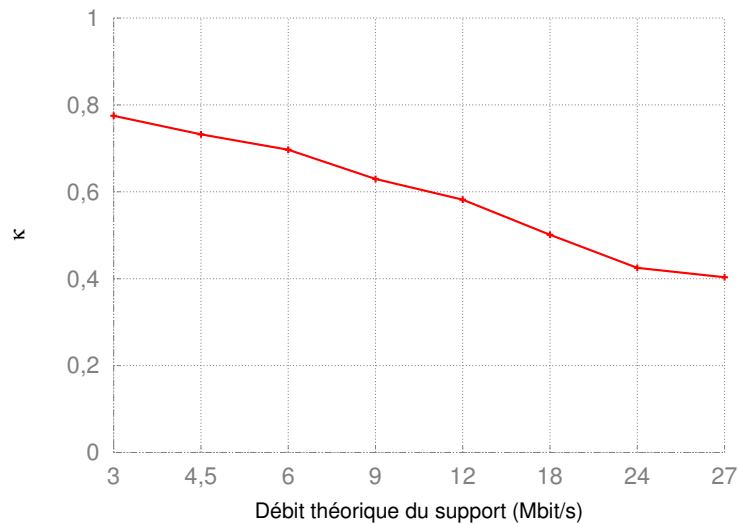


FIGURE 4.6 – κ en fonction du débit (C), la taille du fichier est de 1 KB

Cette première caractérisation montre simplement le surcoût protocolaire (overhead et mécanisme d'accès au médium) induit par les mécanismes mis en jeu. Cela ne correspond donc pas encore complètement à notre système.

La charge ρ_s n'est pas affectée par le nombre des utilisateurs présents dans la zone de portée du point de vue du partage du support, car seul la **RSU** transmet. Les performances du système sont en revanche affectées par le modèle de mobilité, parce qu'au niveau de la couche applicative le débit de données (C_f) est configuré en se servant du nombre moyen de véhicules à portée (N). Le débit C_f reste constant pendant la durée de simulation. Nous allons modéliser la variation de la densité et l'intégrer au modèle de Bianchi.

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

4.4.2 Évaluation de la densité en tenant compte de la mobilité

Pour poursuivre notre étude analytique, nous allons prendre quelques hypothèses. Nous allons tout d'abord supposer, c'est une hypothèse très classique sur autoroute, que les arrivées des véhicules se font selon une loi de Poisson de taux λ . À l'entrée sur le tronçon, la vitesse des véhicules est supposée suivre une loi uniforme entre $[\mu_{min}, \mu_{max}]$; les calculs suivants peuvent être aisément étendus à d'autres lois. Nous supposons enfin que les véhicules vont circuler à vitesse constante sur le tronçon, vitesse choisie à l'arrivée. Nous reviendrons sur l'impact de cette hypothèse par simulation dans les chapitres suivants.

La densité véhiculaire est un paramètre dimensionnant de notre système ; il est donc important de pouvoir le maîtriser dans nos évaluations. Ainsi, nous pourrions évaluer l'impact de la densité sur les performances de l'application, ou au contraire l'impact de certains paramètres à densité constante. Malheureusement, la densité n'est pas un paramètre d'entrée de nos modèles, c'est une conséquence du modèle d'arrivée des véhicules d'une part et du modèle choisi pour leur mobilité de l'autre.

Le modèle de mobilité est également un paramètre important représentant le comportement des véhicules. Nous ne pouvons donc pas l'utiliser afin de maîtriser la densité ; c'est donc la loi d'arrivée des véhicules que nous utiliserons. Cette loi étant une loi de Poisson, c'est le choix de son taux d'arrivée λ qui nous permettra d'atteindre une densité moyenne voulue.

Dans la détermination de λ , la vitesse moyenne, nous venons de le dire, joue un rôle important, il convient donc de l'évaluer correctement.

La première remarque est que la vitesse moyenne en un point ($\bar{\mu}$), par exemple en entrée ou en sortie du tronçon simulé (« time mean speed »), n'est pas la même que sur un tronçon (« space mean speed »). Nous allons estimer la vitesse moyenne des véhicules présents, sachant qu'ils entrent à une vitesse μ_i , pour calculer le taux d'arrivée des véhicules. Nous allons calculer l'espérance du temps de traversée t_i d'un véhicule. Les vitesses en entrée sont choisies ici d'après une distribution uniforme.

$$\begin{aligned}
t_i &= \frac{p}{v_i} \\
E[t_i] &= p \cdot E\left[\frac{1}{\mu_i}\right] \\
E\left[\frac{1}{\mu_i}\right] &= \int_{x=\mu_{min}}^{\mu_{max}} \frac{1}{x} \cdot \frac{1}{(\mu_{max} - \mu_{min})} dx \\
E\left[\frac{1}{\mu_i}\right] &= \frac{\ln(\mu_{max}) - \ln(\mu_{min})}{\mu_{max} - \mu_{min}} \tag{4.4}
\end{aligned}$$

Ce résultat induit le même type de raisonnement que celui utilisé pour montrer la fragilité du modèle random waypoint[60], si $\mu_{min} \rightarrow 0$, le temps de séjour tend aussi vers l'infini. Nous utilisons ici un modèle de mobilité extrêmement simple dans lequel chaque véhicule est caractérisé par une vitesse constante. Cette vitesse est déterminée par une loi uniforme entre μ_{min} et μ_{max} .

Si nous appliquons la formule (4.4) avec les valeurs $\mu_{min} = 32m/s$ et $\mu_{max} = 40m/s$, nous obtenons $\frac{1}{E[\mu_i]} = 0,02769$. Si nous calculons la moyenne temporelle, nous obtenons $E\left[\frac{1}{\mu_i}\right] = 0,02789$. La différence est donc ici minime. Ce résultat était prévisible, car le biais induit par les faibles vitesses est limité du fait que ces dernières prennent leurs valeurs dans un intervalle assez resserré [32 – 40].

4.4.3 Prise en compte de la densité dans le modèle

Chaque OBU va recevoir un débit applicatif constant égal à $C_f = \frac{C_a}{N}$. L'overhead (en-têtes + méthode d'accès) occupe $(1 - \kappa) = 22,5\%$ du support. Le débit équivalent à cette charge donne $C_{fo} = C_f \cdot (1/\kappa)$. Le support est donc convoité pendant $\frac{N \cdot C_{fo}}{C_a} = \frac{1}{\kappa} \sim \frac{1}{0,775} = 1,29$ du temps. Comme au mieux, il ne peut être occupé que 100% du temps le taux de perte de paquets vaut a minima $(1 - \kappa)$.

Tenons compte désormais du nombre variable de véhicules. La zone de couverture se modélise comme une file $M/G/\infty$ avec des arrivées poissonniennes de taux λ (le débit d'arrivée des véhicules en veh/h) et de temps de séjour t_i . En régime permanent, le nombre de clients n dans la file suit une loi de Poisson de paramètre $\lambda \cdot E[t_i] = N$.

$$P_r[n = i] = \frac{N^i}{i!} e^{-N} \tag{4.5}$$

$$E[n] = \sum_{i=0}^{\infty} i \frac{N^i}{i!} e^{-N} = N \tag{4.6}$$

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

S'il existe i flux actifs, chacun reçoit un débit applicatif C_f , ce qui correspond à un débit C_{fo} sur le support. Dès que le débit $i \cdot C_{fo} > C$ le trafic en excès est rejeté. Le trafic émis est donc égal à $\min(i \cdot C_{fo}, C)$ si nous avons i flux actifs. Finalement, le débit réellement émis (Θ) sur le support vaut $\sum_{i=0}^{\infty} (\min(i \cdot C_{fo}, C)) \cdot P_r$ soit encore

$$\Theta = \sum_{i=0}^{\infty} (\min(i \cdot C_{fo}, C)) \cdot \frac{N^i}{i!} e^{-N} \quad (4.7)$$

Ce modèle est en fait un modèle fluide qui consiste à considérer qu'en deçà d'une charge normalisée de 1, le trafic est complètement écoulé. Les motivations proviennent des faibles tailles des buffers et du fait que le trafic soit du type CBR.

Le trafic utile est donc $\rho_s = \frac{\Theta \cdot k}{C}$. La figure 4.7 montre les différentes valeurs de charge normalisée en fonction de la densité N_k , pour les trois scénarios.

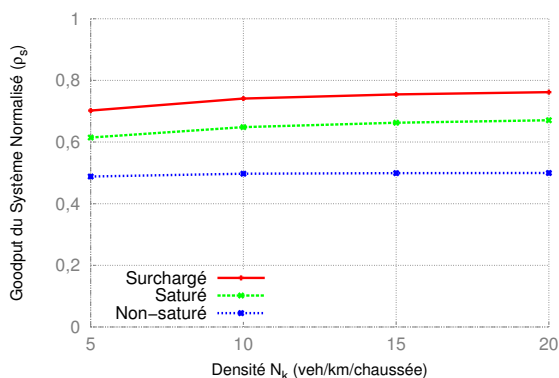


FIGURE 4.7 – Modèle de Bianchi intégrant l'impact de la variation de la densité

4.5 Paramètres de simulation

Pour conforter ces premiers résultats analytiques, nous avons mis en œuvre des simulations. Le modèle Three Log Distance nous a permis de calculer la portée de la RSU à 594,1 m. Nous utilisons $\mu_{min} = 32 \text{ m/s}$ et $\mu_{max} = 40 \text{ m/s}$. Nous avons choisi ces vitesses en supposant que la plupart de conducteurs vont vouloir rouler le plus vite possible. Dans ces premières simulations, la vitesse des véhicules ne change plus, une fois qu'ils sont entrés sur le tronçon d'autoroute. Les véhicules vont rouler autour de la vitesse moyenne qui est approximativement 130 km/h, la vitesse maxi-

male autorisée sur une autoroute française. Nous considérons un ensemble de densités, $\{5, 10, 15, 20\}$ *veh/km/chaussée*.

Les paramètres de la norme 802.11p ont été utilisés pour configurer la couche physique dans le simulateur. Les simulations ont été réalisées avec un débit du support $C = 3\text{Mbit/s}$ et un seuil de sensibilité de -85dBm [59, Table 17-13]. La méthode d'accès au support employée est le RTS/CTS. L'application peut utiliser différents canaux de service. [8, I.2.2] montre les différentes spécifications pour chaque classe de canal. Nous allons utiliser l'EIRP maximal (23dBm) avec un gain en réception de 1dBi .

La charge utile pour chacun des paquets est fixée à un kilo-octet. En ajoutant les en-têtes, la longueur totale de PSDU représente 1088 octets³.

Le simulateur utilisé est NS3 version 10. Chaque simulation correspond à 1000 secondes du fonctionnement du système. Chaque scénario a été simulé dix fois. Les intervalles de confiance sont présentés pour un niveau de confiance de 95%. Le simulateur applique le modèle « Three Log Distance » pour caractériser l'affaiblissement de propagation.

En nous fondant sur les résultats obtenus à partir du scénario surchargé, nous avons retenu $\rho_e = 0,70$ comme paramètre d'entrée pour le deuxième scénario qui utilise la capacité maximale. Pour le dernier scénario, nous avons arbitrairement fixé $\rho_e = 0,50$. Ainsi, la couche applicative envoie avec un débit agrégé de $C_a = C \cdot 0,70$ et de $C_a = C \cdot 0,50$ respectivement.

4.6 Résultats

Dans cette section, nous allons présenter des résultats de simulation obtenus avec les 3 charges de trafic. Nous allons nous concentrer sur le Goodput du système, le PDR, le délai, l'équité et, finalement la fragmentation. Ce dernier indicateur est pertinent, car une application, même si elle n'a pas forcément besoin d'un fichier complet, a besoin d'une certaine continuité de ses fragments.

4.6.1 Le Goodput du système

La charge normalisée obtenue par simulation montre la relation entre le débit de la charge utile reçue par l'ensemble des véhicules à portée de la RSU, et le débit théorique

3. La norme DSRC[61] mentionne que le dispositif doit être capable de transférer des messages avec un taux d'erreur de paquet inférieur à 10% pour des véhicules roulant à une vitesse de 85mph (136km/h) en utilisant une longueur de PSDU de 1000 octets. Comme nous l'avons mentionné auparavant la vitesse moyenne utilisée en entrée de la portion d'autoroute simulée est de 130km/h .

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

du support.

La figure 4.8 représente la charge normalisée ρ_s en fonction de la densité N_k pour les trois scénarios, calculée par le modèle analytique et par simulation. Nous pouvons observer que les résultats du modèle analytique sont confortés par les résultats de simulations.

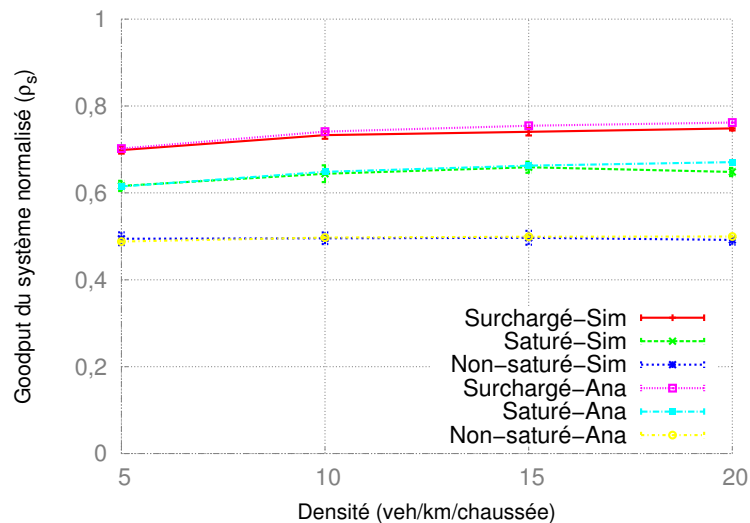


FIGURE 4.8 – Goodput du système normalisé

4.6.2 Taux de délivrance de paquets

Il est évident que lorsque le système est surchargé, les paquets envoyés par le serveur ne seront pas tous reçus par les OBU. La figure 4.9 illustre l'impact de chacun des scénarios sur le PDR. Dans le premier scénario, les OBU reçoivent une moyenne de 75% de paquets, cela signifie un « Taux de perte de paquets (PLR) » de 25%. Le deuxième scénario vise à minimiser le PLR, tout en optimisant l'utilisation du support. De bons résultats sont obtenus dans ce scénario. Cependant, il existe toujours une PLR d'environ 10%. Le dernier scénario montre un PDR proche de 100%, mais le PLR persiste. Le problème est lié à la densité instantanée. Quand la charge d'entrée est faible, l'influence de la mobilité s'estompe.

Nous avons mis en évidence qu'avoir un débit constant (C_f) pour chaque véhicule (flux unicast) n'est pas efficace. Le mieux consiste à avoir un débit constant pour l'ensemble des véhicules (C_a) et d'avoir un mécanisme d'ordonnancement pour envoyer les paquets en fonction du nombre de véhicules à portée. Par exemple, l'application

pourrait utiliser l'algorithme d'ordonnancement Round-Robin, de cette manière avec l'application proposée, C_a serait constant tout au long de la simulation.

Dans le cas où le support est partagé, la **RSU** et les **OBU**s étant en compétition pour l'accès au support, un mécanisme d'adaptation de C_a pourrait être envisagé. C_a serait calculé en fonction de la disponibilité de bande passante sur le support. Cette disponibilité serait obtenue depuis la **RSU**.

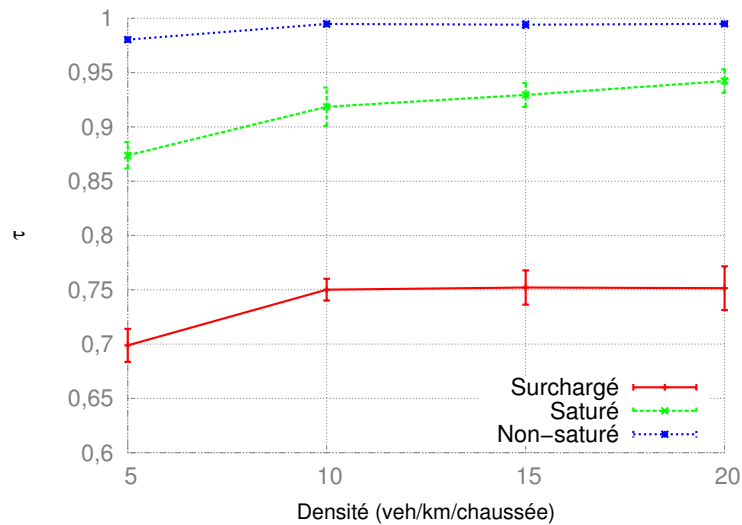


FIGURE 4.9 – Taux de délivrance de paquets en fonction de la densité de véhicules

4.6.3 Délai

La figure 4.10 illustre l'impact de la *densité du trafic de données* sur le délai (chaque scénario a été choisi avec une charge de trafic spécifique). Il est donc important de trouver une valeur maximale de la *densité du trafic de données*, ce qui permettrait d'exploiter suffisamment le support sans fil avec une qualité de service acceptable. Un système surchargé souffre d'un délai important, ce délai étant fonction de la taille de la file d'attente. Cependant, l'impact de ce délai sur une application de téléchargement de contenus est négligeable.

Selon les résultats, il y a un impact négligeable de la *densité de trafic véhiculaire* sur le délai d'un scénario. Cela est dû au fait que le débit des flux est adapté en fonction de cette densité ($C_f = C_a/N$).

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

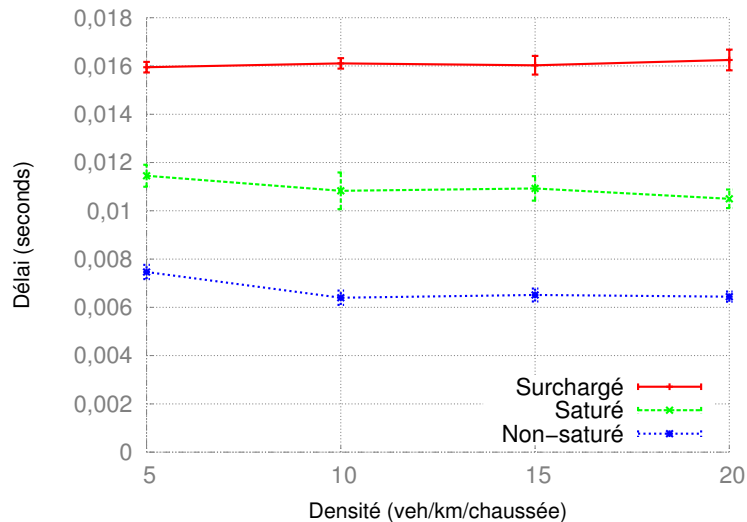


FIGURE 4.10 – Délai en fonction de la densité de véhicules

4.6.4 Équité

Comme le montre la figure 4.11, l'équité n'est pas affectée de manière importante par la densité véhiculaire. Les résultats sont très bons pour l'ensemble des scénarios. Quoiqu'il en soit, le système est moins équitable lors de l'utilisation d'un système surchargé. L'application envoie des paquets avec un débit constant vers les OBUs. L'iniquité est introduite lorsque les paquets sont perdus dans la file d'attente surchargée de la RSU. Des mécanismes fondés sur l'équité pour gérer les pertes sont absents de la file d'attente. Lorsque la file d'attente est remplie, les paquets entrants sont rejetés.

Le troisième scénario n'est pas parfaitement équitable, même si peu de paquets ont été éliminés de la file d'attente de la RSU. L'iniquité, dans ce cas, est une conséquence directe de la mobilité. Si une OBU est plus rapide, elle a moins de temps pour télécharger les données. L'équité est donc affectée par la perte de paquets et par la mobilité des OBUs.

4.6.5 Fragmentation

Les applications exigent généralement que le fichier soit complet pour pouvoir l'utiliser. D'autres applications, telles que la mise à jour de cartes, peuvent utiliser certains fragments de fichier pour fonctionner même si le fichier n'est pas complet. Cependant, cela peut nécessiter un certain nombre de morceaux contigus du fichier.

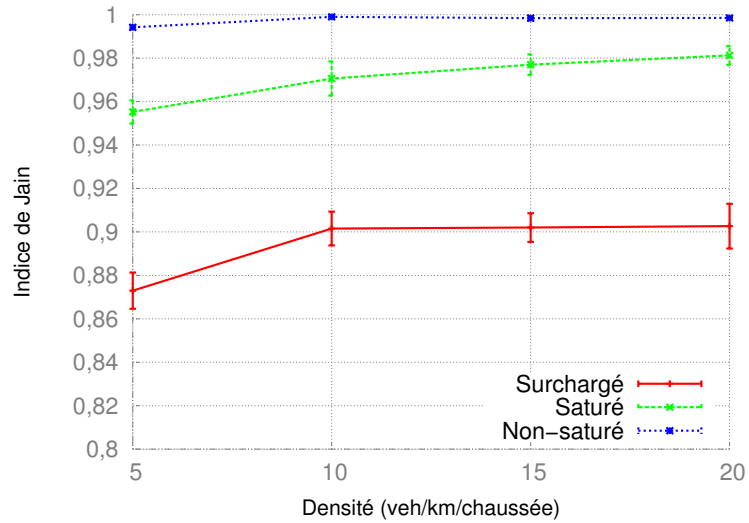


FIGURE 4.11 – L'équité en fonction de la densité de véhicules

Le niveau de fragmentation d'un fichier est donné par la métrique de fragmentation (Cette dernière va de 0, quand le fichier est très fragmenté, à 1, quand il ne l'est pas du tout). La figure 4.12 montre les indices de fragmentation pour chaque scénario. Elle montre également les courbes optimales pour lesquelles tous les fragments reçus sont contigus, mais le fichier n'est pas complet. De cette façon, nous pouvons avoir une meilleure perception de la fragmentation compte tenu du taux de perte de paquets. À cet égard, le taux est grandement affecté par la *densité du trafic de données*.

4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE CONTENUS : CONTEXTE AUTOROUTIER

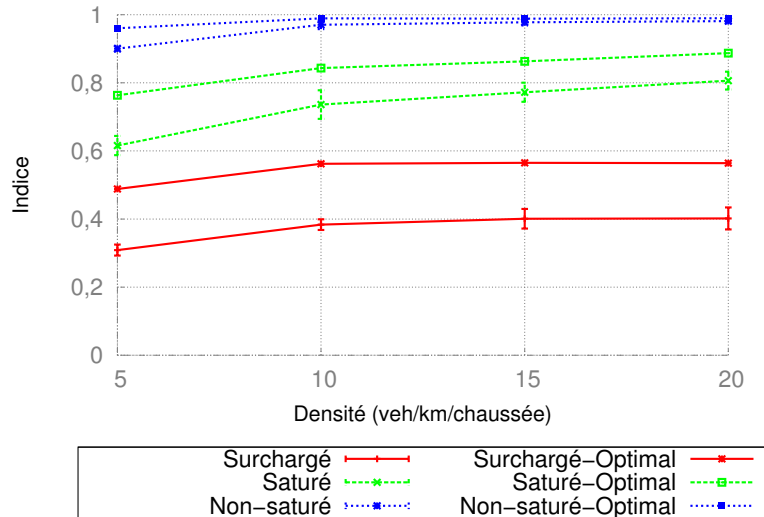


FIGURE 4.12 – Fragmentation en fonction de la densité de véhicules

4.7 Conclusions

Les critères présentés sont importants pour une bonne appréciation du comportement de l'application. En fait, dans un système surchargé, les performances ne sont pas acceptables. Ces résultats suggèrent l'importance d'une interaction entre la [RSU](#) et le serveur, afin de gérer de façon efficace le débit disponible dans la [RSU](#), et pouvoir maintenir une qualité de service acceptable.

La plupart des applications utilisent le service réseau du type « best effort ». Cependant, une application de mise à jour de cartes comporte un élément supplémentaire, cet élément est l'équité. Ceci pourrait être géré par le serveur d'applications à l'aide d'une méthode d'ordonnancement pour envoyer la même quantité de données à chaque véhicule par le biais de la [RSU](#).

L'ordonnanceur pourrait se servir d'informations supplémentaires fournies par la [RSU](#) ou par les [OBUs](#) : par exemple le débit disponible dans la [RSU](#) pour maximiser l'utilisation du support, ou les vitesses des véhicules pour améliorer l'équité. En outre, il faut prendre en compte que plus les [OBUs](#) envoient des informations de contrôle vers le serveur, moins la bande passante sera disponible pour la charge utile. Il faut donc trouver un compromis.

Le modèle de Bianchi induit une surestimation de la charge de sortie ρ_s (Goodput du système Normalisé) par rapport à celle obtenue par simulation. Cela provient du fait

que le serveur prend comme paramètre la densité véhiculaire qui pour lui sera constante tout au long de la simulation, alors que la densité réelle change (modèle de mobilité).

Nous avons modélisé le Goodput du système en tenant compte de la mobilité, à partir d'une file $M/G/\infty$, les résultats du modèle confortent ceux des simulations.

Nous constatons qu'utiliser un débit constant par véhicule est une bonne démarche à faible densité (scénario en sous-charge). Cependant, dès que la densité dépasse un certain seuil, les résultats deviennent très mauvais (système surchargé). Donc, cette approche ne tient pas le facteur d'échelle.

Si nous avons la volonté d'avoir un densité plus importante, il faut que le débit puisse être adapté. Cette adaptation se fera fonction de la densité de véhicules dans la portée, et de la disponibilité de la bande passante d'une [RSU](#). Une approche pouvant aider à cette démarche est l'utilisation d'algorithmes d'ordonnancement, par exemple « Round-Robin ». L'application pourrait adapter C_a en fonction de la bande passante disponible dans la [RSU](#), et l'algorithme pourrait partager de façon équitable C_a en fonction du nombre de véhicules. Cette approche est utilisée dans le chapitre suivant pour analyser le facteur d'échelle.

Il faut retenir surtout que l'application envisagée est la mise à jour. Celle-ci cherche à envoyer un fichier de mise à jour vers tous les véhicules. Une application en unicast n'est pas très probablement la solution la plus adaptée.

**4. COMMUNICATIONS I2V POUR LA DISTRIBUTION DE
CONTENUS : CONTEXTE AUTOROUTIER**

Chapitre 5

Passage à l'échelle - Comparaison unicast multicast

5.1 Description du problème

Dans ce chapitre, nous poursuivons notre analyse en considérant cette fois deux schémas de délivrance : unicast et multicast, en utilisant dans un premier temps une seule [RSU](#). Le scénario est étendu ensuite à six [RSUs](#). L'idée est alors d'étudier l'impact des intermittences sur le téléchargement de contenus. Un des points qui est apparu comme majeur dans le cas du multicast est la « Stratégie d'ordonnancement ». Quatre solutions ont été mises en œuvre.

Les études ont été menées dans ce chapitre à l'aide de simulations que l'on a voulues plus réalistes que dans le chapitre précédent. Nous avons également mis en place des modèles mathématiques dans certains cas ; ils permettent en effet de mieux comprendre les phénomènes en présence et leur importance relative.

5.2 Modèle du système

5.2.1 Modèle de mobilité

Le schéma général du scénario a été présenté dans le chapitre 4, il est rappelé dans la figure 5.1. La longueur (L) de l'autoroute est de douze kilomètres. Le modèle de mobilité exploité dans ce chapitre est l'IDM/MOBIL[62, 63]. Ce dernier est un modèle réaliste du type microscopique. Dans le chapitre précédent, nous avons constaté la faiblesse des solutions unicast. Ces faiblesses seront encore présentes dans le multi-[RSUs](#)

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

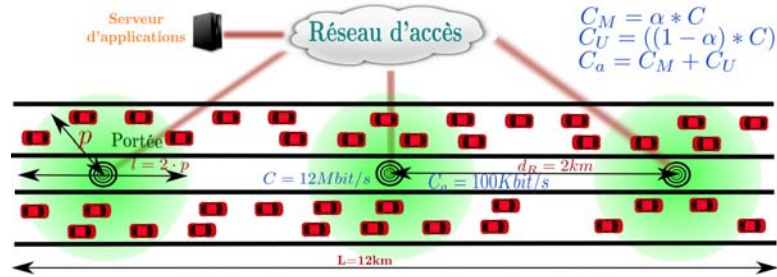


FIGURE 5.1 – Scénario

et par conséquent, ce chapitre a pour vocation essentielle d’analyser les performances du multicast. Dans ce contexte, la densité de véhicules a un impact moindre sur l’application. Pour limiter le nombre de paramètres qui varieront, nous allons fixer la densité de véhicules.

Le taux d’entrée de véhicules sera égal à $\lambda = 0,3 \text{ veh/s}$, avec des arrivées Poissonniennes. La vitesse du véhicule à l’entrée de l’autoroute suit une loi normale de moyenne $\bar{\mu} = 36 \text{ m/s}$ et un écart-type de $\sigma_{\mu} = 1 \text{ m/s}$.

Notons toutefois, un effet secondaire lors de l’utilisation de ce modèle qui est l’impact direct de la densité sur la vitesse moyenne des véhicules. Un taux élevé d’entrée mène à une faible vitesse (en raison des embouteillages). Évidemment, un taux inférieur influe moins sur la vitesse du trafic, la vitesse du véhicule étant alors essentiellement liée au comportement individuel du conducteur.

Cette propriété complique l’analyse, car si l’intensité du trafic entrant est un paramètre de notre problème, la densité de véhicules sur l’autoroute n’est pas facilement calculable même si l’on se donne la loi du vecteur vitesse à l’entrée de l’autoroute. Ce phénomène aura une influence sur les résultats, que nous tâcherons d’évaluer.

5.2.2 La topologie réseau

La topologie est composée par le serveur d’applications, le réseau d’accès, les **RSUs** et finalement les **OBUs**. Dans le serveur est installée l’application pour fournir les contenus aux **OBUs** (figure 5.1).

Nous allons utiliser deux scénarios d’autoroute. Un scénario avec une unique **RSU** et un avec six **RSUs**. Dans le premier scénario, la **RSU** se trouve au milieu de l’autoroute. Dans le deuxième, les **RSUs** sont régulièrement réparties le long de l’autoroute, tous les deux kilomètres.

Certains paramètres ont été ajustés de façon optimiste (*p. ex.*, tous les véhicules

sont des berlines avec une unité de communication intégrée, les vitesses sont choisies en fonction des normes françaises, ...).

5.2.3 Le modèle applicatif

Une application serveur fournit aux véhicules la carte (ou toute autre donnée utile). La carte est divisée en morceaux notés m_i de taille constante, chaque morceau est envoyé dans un seul paquet. Nous notons M la taille du fichier en nombre de morceaux.

Nous supposons que la carte ne subit pas de mises à jour au cours de la période de mesure, et donc la version de la carte envoyée périodiquement sera la même. L'application dispose de deux schémas de délivrance (unicast, multicast).

5.2.3.1 L'application en unicast

En utilisant l'unicast, le client envoie une requête en demandant le fichier. La communication est établie de bout en bout et l'application doit choisir le véhicule et le morceau de fichier à envoyer. Nous avons mis en place un ordonnanceur simple fondé sur la méthode round-robin pour contrôler l'équité tout en optimisant la bande passante disponible. De nombreuses autres solutions auraient été envisageables pour favoriser par exemple les utilisateurs ayant reçu une grande partie du fichier et améliorer par conséquent la proportion de véhicules ayant téléchargé l'intégralité du fichier. Comme nous le verrons par la suite, l'unicast seul conduit à des performances médiocres et c'est pour cela que dans ce chapitre, nous nous sommes contentés de cette méthode simple. Vu les piètres performances de l'unicast seul, nous ne traiterons pas non plus le multi-**RSUs**. On peut simplement supposer que quand un véhicule entre à portée d'une nouvelle **RSU**, il reprend le téléchargement là où il s'en est arrêté. L'application possède une liste de véhicules qui sont dans la couverture de la **RSU**.

5.2.3.2 L'application en multicast

Dans le scénario avec une seule **OBU**, nous avons utilisé un seul flux multicast entre le serveur et l'**OBU**. Pour le scénario avec plusieurs **RSUs**, différentes configurations de flux entre le serveur et les **OBUs** sont possibles. Dans la première, un seul flux multicast est considéré pour toutes les **OBUs** (cf. section 5.6.2). Dans la seconde configuration, un flux multicast indépendant pour chaque **RSU** est utilisé. Nous allons supposer que les **RSUs** n'ont pas de capacité de stockage (sauf la file d'attente d'émission). Chaque paquet arrivant à la **RSU** doit être relayé vers les **OBUs** qui sont dans la couverture de la **RSU**.

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

Avec les configurations mentionnées ci-dessus, la bande passante sur le réseau d'accès est utilisée en permanence. Pour pallier ce problème, l'idéal serait d'avoir une capacité de stockage dans les **RSUs**, de sorte que la **RSU** devienne un serveur cache. Ainsi, le serveur pourrait déléguer les transmissions périodiques de fichiers aux **RSUs**. Dans notre étude, nous n'avons utilisé que les deux premières configurations, l'implantation d'un cache dans la **RSU** nous semble peu probable, car trop coûteux.

Nous considérons que les **RSUs** consacrent un débit constant à l'application. Les flux multicast auront un débit constant au niveau de la couche applicative de $C_a = 100 \text{ Kbit/s}$.

Nous avons défini deux stratégies d'envoi de morceaux de fichier. La « Stratégie d'ordonnancement Séquentielle », qui est la plus simple, consiste à transmettre des morceaux dans l'ordre où ils apparaissent dans le fichier. La seconde sera la « Stratégie d'ordonnancement Aléatoire » du type uniforme sans remise et différera pour chaque **RSU**.

Le serveur envoie un fichier indéfiniment comme une séquence de M morceaux, m_0, \dots, m_{M-1} . L'ordre dans lequel les morceaux sont envoyés à travers la **RSU** R_i est donné par la permutation $\sigma_{R_i} : \{0 \dots (M-1)\} \mapsto \{0 \dots (M-1)\}$.

En utilisant la stratégie séquentielle, cela revient à considérer $\sigma_{R_i} = Id_{\{0 \dots (M-1)\}}$ pour tout R_i , en utilisant la technique aléatoire, chaque σ_{R_i} sera une permutation aléatoire potentiellement différente.

Quand une **OBU** traverse plusieurs aires couvertes par des **RSUs**, elle peut alors recevoir des doublons. Ce phénomène va se révéler encore plus crucial que la perte de paquets. En outre, la durée de téléchargement s'en trouve allongée.

5.2.3.3 Combinaison de l'unicast et du multicast

L'objectif de cette combinaison consiste à traiter les problèmes de doublon. Notre idée est que les phénomènes de doublon risquent de conduire à des situations dans lesquelles tous les véhicules possèdent une grande partie de la carte, mais que peu ont la totalité de la carte, que ces morceaux manquants sont distincts et pas dans la liste des prochains envois.

Un véhicule entrant dans l'aire d'une **RSU** peut envoyer au serveur un bitmap qui contient les morceaux manquants. Les véhicules seront supposés recevoir le flux en unicast ou en multicast pendant qu'ils sont couverts par une **RSU**. Pour effectuer des comparaisons équitables, nous proposons dans cette combinaison de consacrer un débit total agrégé identique à celui du multicast seul. Il apparaît instantanément que la partie

consacrée à l'unicast doit être faible (tous les véhicules étant supposés intéressés par les flux). Nous avons décidé de fonder le choix entre l'unicast et le multicast sur le taux de réception τ du véhicule : nombre de morceaux reçus divisé par la taille du fichier en paquets M . Si $\tau > \tau_s$ (où τ_s est un seuil à définir), une OBU est autorisée à envoyer son bitmap, une transmission en unicast est ensuite utilisée pour ce véhicule.

La question suivante est la répartition du débit disponible entre les véhicules. Si n_U est le nombre de véhicules utilisant le schéma unicast dans la zone de portée d'une RSU, et n_M ceux qui utilisent le multicast, on peut définir $\alpha = n_M / (n_M + n_U)$. Si C_a est la bande passante disponible pour l'application dans cette RSU, la stratégie d'ordonnement choisira un débit $C_M = \alpha \cdot C_a$ pour le multicast, et une répartition équitable des $C_U = (1 - \alpha) \cdot C_a$ pour les n_U transmissions en unicast. Les morceaux manquants envoyés en unicast le sont de façon séquentielle. On suppose donc que le serveur connaît le nombre de nœuds qu'il sert et que donc de façon implicite ou explicite, il sait quand le véhicule entre à portée de la RSU et quand il la quitte.

5.2.3.4 Le morceau le plus recherché en premier

Nous allons analyser la performance quand l'application utilise des informations de feedback. Ces informations sont envoyées par les OBUs vers la RSU. L'information envoyée est un vecteur bitmap a^n . Ce bitmap contient l'état de chaque morceau (reçu ou pas) $a^n = \{a_0^n, \dots, a_{M-1}^n\}$ où $a_k^n = 1$ si l'OBU n possède le morceau m_k , et $a_k^n = 0$ dans le cas contraire. L'OBU envoie son bitmap quand elle rentre dans la couverture de la RSU. La RSU maintient un ensemble de bitmaps $\mathbf{B} = \{a^0, a^1, \dots, a^{N-1}\}$ où N est le nombre constant de véhicules à portée de la RSU (cf. figure 5.2). On maintient également un vecteur $\mathbf{f} = (f_0, \dots, f_{M-1})$ où f_k est le nombre de véhicules à portée n'ayant pas le morceau m_k . Ces nombres sont mis à jour lors de l'arrivée ou le départ des véhicules ainsi qu'au fur et à mesure des envois successifs.

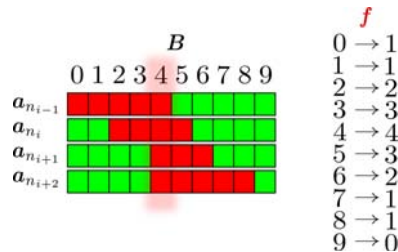


FIGURE 5.2 – Vecteur de bitmaps et vecteur de fréquences

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

L'application choisit d'envoyer le morceau le plus manquant m_k . L'envoi effectué, f_k est mis à zéro et les bitmaps des nœuds utilisant le multicast sont mis à jour dans le serveur en supposant les transmissions correctes (figure 5.3). On ne met bien sûr pas en place de mécanisme d'accusés de réception.

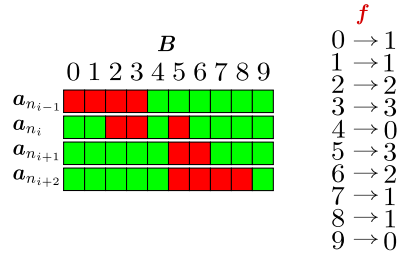


FIGURE 5.3 – Vecteur de bitmaps et vecteur de fréquences après envoi du morceau 4

5.3 Métriques de performances

Nous allons évaluer le nombre de morceaux reçus par chaque OBU à la fin de son passage dans le tronçon d'autoroute dans ces différentes configurations. Ce critère de performance peut être mesuré par plusieurs indicateurs, nous allons nous concentrer sur les deux suivants :

- La proportion de téléchargement complet π_1 nous aidera à estimer la proportion de clients satisfaits.
- Le taux moyen de réception (τ) peut être une estimation moyenne de satisfaction du client, si un téléchargement partiel est utile (par exemple, dans le téléchargement de mises à jour de cartes, une partie de la mise à jour peut être utilisée).

5.4 Paramètres de simulation

Le simulateur utilisé est NS3 version 11. Chaque simulation correspond à un temps simulé de 3000 secondes. Chaque résultat a été obtenu avec 10 simulations. L'intervalle de confiance montré dans les figures obtenues par simulation correspond à un niveau de confiance de 95%.

Le modèle NIST[64] est utilisé pour simuler le taux d'erreur. « Three Log Distance » modélise l'affaiblissement de propagation et « Nakagami » la variation de l'atténuation (fading).

La couche physique a été paramétrée avec les valeurs du 802.11p. Les simulations ont été réalisées avec un débit du support de 12 Mbit/s et un seuil de sensibilité de -77dBm [59, Table 17-13]. La méthode d'accès au support utilisée dans la **RSU** pour l'unicast est le RTS/CTS, le multicast utilise uniquement le backoff minimal.

Les architectures de communication définissent plusieurs canaux. Dans le profil européen, les deux premiers canaux sont dédiés à des applications de sécurité routière et d'efficacité. Dans cette étude, nous utilisons le second dénommé G5SC2. Les principales caractéristiques (débit 12 Mbit/s , et puissance d'émission limitée à 23dBm EIRP) ont été définies dans notre modèle de simulation.

La charge utile pour chacun des paquets est fixé à un kilo-octet. En ajoutant les en-têtes, cela fait un total de 1088 octets pour la **PSDU**. Les spécifications **DSRC**[61] mentionnent que le dispositif doit être capable de transférer des messages avec un taux d'erreur de paquet inférieur à 10% pour des véhicules roulant à une vitesse de 85 mph (136 km/h) en utilisant une longueur de **PSDU** de 1000 octets.

5.5 Modèle analytique de la stratégie d'ordonnancement aléatoire sur un système multicast

Dans un système utilisant le multicast pur, le comportement de la stratégie d'ordonnancement aléatoire peut être modélisé simplement de sorte que la performance puisse être calculée analytiquement. Les seules restrictions par rapport au modèle utilisé pour les simulations sont que l'on suppose une vitesse constante pour un véhicule donné, et une répartition uniforme des pertes dues à des erreurs de transmission au sein de la zone de réception, avec une probabilité d'erreur de paquet p_e . Nous aurions pu relâcher la contrainte de la vitesse constante, car finalement ce qui nous importera sera la loi de la durée de couverture du véhicule par la **RSU**. En revanche, les ralentissements dus aux autres véhicules sont très difficiles à modéliser.

Appelons M_R le nombre de morceaux de fichiers que l'**OBU** peut, théoriquement, recevoir dans une seule **RSU**. Cette valeur est fonction de la vitesse du véhicule et du débit. M_R sera donc ici une constante. Le paramètre M_R dépend du temps de traversée du véhicule. Nous supposons que la taille du fichier M est plus grande que M_R .

5.5.1 Analyse du nombre de nouvelles transmissions

Si M_O est le nombre de morceaux de fichier reçus par une **OBU** quand elle rentre à portée d'une **RSU**, la probabilité qu'elle reçoive k nouveaux morceaux à l'intérieur de

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

sa zone de couverture, est donnée par la distribution hypergéométrique :

$$HG(M, M - M_O, M_R, k) = \binom{M - M_O}{k} \cdot \binom{M_O}{M_R - k} / \binom{M}{M_R}$$

Il faut bien évidemment faire attention aux bornes des valeurs pouvant être prises par M_O et $(M_R - k)$.

5.5.2 Analyse du nombre de nouvelles réceptions

La probabilité qu'une **OBU** reçoive correctement q nouveaux morceaux, parmi k qui lui sont envoyés, est donnée par :

$$\pi_{k,q} = \binom{k}{q} \cdot p_e^{k-q} \cdot (1 - p_e)^q$$

Par conséquent, la probabilité qu'une **OBU** reçoive q nouveaux morceaux à travers la zone de transmission d'une **RSU** est :

$$\Gamma_{M, M_O, M_R, q} = \sum_{k=q}^n HG(M, M - M_O, M_R, k) \cdot \pi_{k,q}$$

5.5.3 Analyse complète avec plusieurs RSUs

Soit $P_i(b)$ la probabilité qu'un véhicule reçoive i morceaux différents après avoir traversé b **RSUs** consécutives. En utilisant la définition ci-dessus de Γ , nous avons :

$$P_i(b) = \sum_{j=\max(0, i-n)}^i \Gamma_{M, j, n, i-j} \cdot P_j(b-1)$$

Soit $P(b)$ le vecteur de $P_i(b)$ et $G(M, M_R)$ la matrice des $\Gamma_{M, j, M_R, i-j}$, nous avons :

$$P(b+1) = G(M, M_R) \cdot P(b)$$

En supposant que $P(0) = 0$, nous pouvons calculer de manière itérative le taux de réception après un certain nombre de **RSUs**. Nous allons comparer ces résultats avec ceux obtenus par simulation (cf. figure 5.8).

5.6 Résultats de simulation

Avec les paramètres ci-dessus, chaque **RSU** peut transmettre à une distance allant jusqu'à 792 mètres. Le nombre maximum de morceaux de fichier qu'un véhicule peut (théoriquement) recevoir d'une seule **RSU** est de $M_R = 275$. Nous allons l'utiliser comme valeur de référence pour la taille de la carte dans les simulations.

5.6.1 Multicast vs unicast avec une seule RSU

Dans cette section, nous utiliserons le premier scénario d'autoroute (5.2.1). La taille du fichier utilisée est $M = M_R$. La figure 5.4 montre le taux moyen de réception à la fois pour un schéma multicast pur ($\tau_s = 0$) et pour un schéma unicast ($\tau_s = 1$).

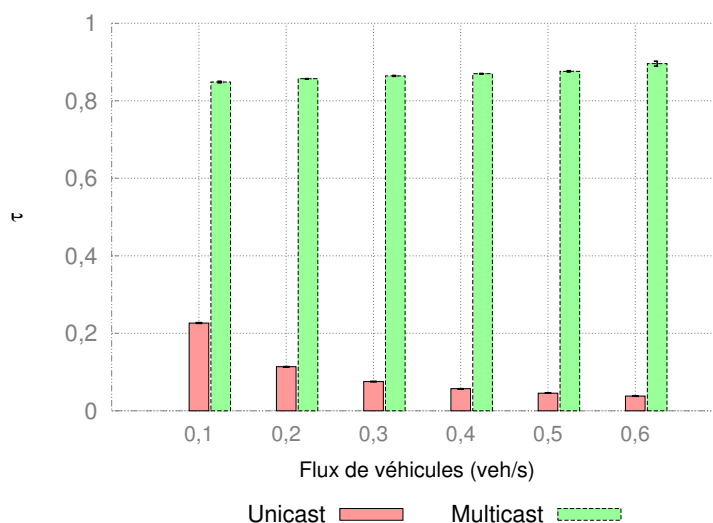


FIGURE 5.4 – Taux de réception τ en fonction du flux véhiculaire

On constate donc que l'unicast conduit à des performances catastrophiques puisque même à densité véhiculaire faible, le taux de réception est quatre fois inférieure à celui obtenu avec le multicast. Qui plus est, ce taux de réception décroît avec la densité de véhicules ; ce mécanisme ne passe évidemment pas à l'échelle.

5.6.2 Le comportement de « multicast » sur un scénario avec plusieurs RSU

Depuis cette section, le deuxième scénario d'autoroute (5.2.1) est utilisé. Nous allons évaluer la performance d'un système utilisant multicast pur ($\tau_s = 0$).

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

Dans ces solutions multi-*RSUs*, nous allons augmenter la taille des fichiers pour tester les limites de techniques multicast. Les solutions « unicast pur » n'ont alors plus aucun intérêt. En revanche, la stratégie d'ordonnancement multicast retrouve toute son importance.

5.6.2.1 Stratégie d'ordonnancement Séquentielle

Les figures 5.5 et 5.6 montrent, pour différentes tailles de carte, le taux de réception moyen au long de plusieurs *RSUs* en utilisant la Stratégie d'ordonnancement Séquentielle. Le taux de perte de paquets est d'environ 15%. La figure 5.5 montre le taux de réception moyen cumulé après avoir traversé toutes les *RSUs* avec les intervalles de confiance et la figure 5.6 le taux de réception moyen cumulé dégroupé par *RSU*.

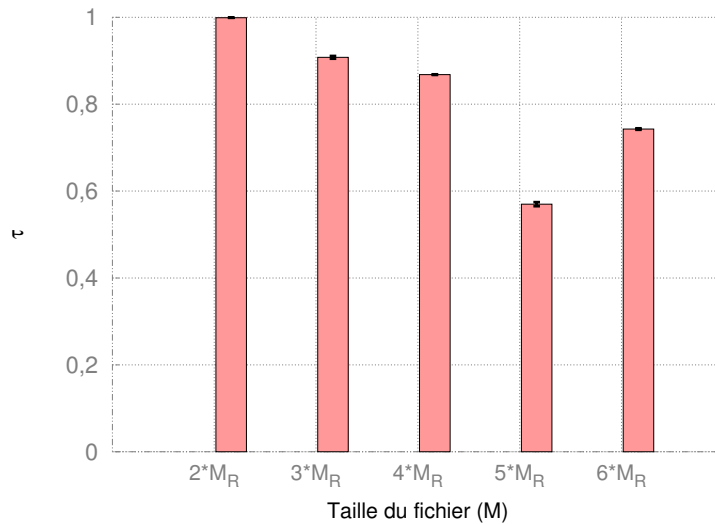


FIGURE 5.5 – τ en fonction de M avec la Stratégie Séquentielle

Nous pouvons observer une valeur étonnamment faible pour $M = 5 \cdot M_R$. Le problème est précisément qu'avec cette configuration, de nombreux véhicules subissent un phénomène de synchronisation : à partir de la troisième *RSU* (dans ce cas), la plupart des morceaux reçus ont été déjà reçus auparavant.

La Stratégie devient donc inéquitable. Quelle que soit la configuration, certains véhicules, en fonction de leur vitesse et d'autres paramètres, peuvent recevoir beaucoup moins de morceaux que les autres.

Quant à la proportion d'*OBUs* qui ont téléchargé une carte complète, π_1 , le tableau 5.1 montre la valeur en fonction de la taille du fichier (M) et du nombre de *RSUs*.

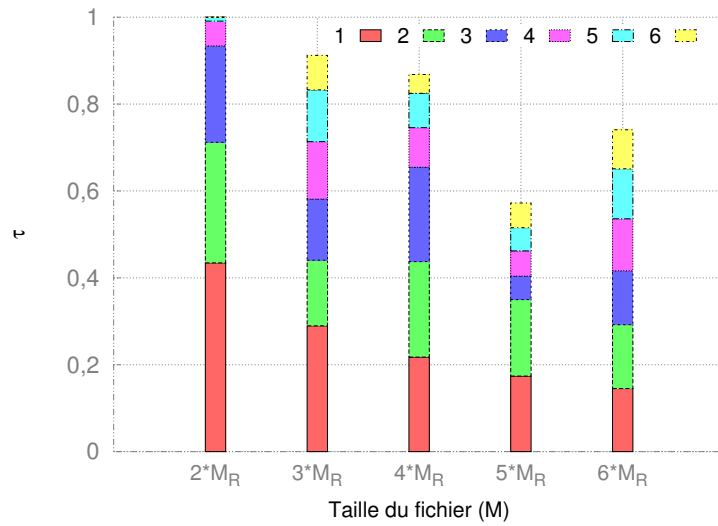


FIGURE 5.6 – τ en fonction de M avec la Stratégie Séquentielle, τ est dégroupé par RSU

TABLE 5.1 – Taux π_1 des OBU_s ayant le fichier complet en utilisant la Stratégie Séquentielle en sortant de la portée des différentes RSU_s

M	numéro de la RSU						Total
	1	2	3	4	5	6	
1 * M_R	0,29	27,81	65,07	6,43	0,39	0,01	100
2 * M_R	0,0	0,0	0,43	6,34	29,44	34,50	70,71
3 * M_R	0,0	0,0	0,0	0,0	0,0	0,74	0,74

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

5.6.2.2 Stratégie d'ordonnancement Aléatoire

Les figures 5.7 et 5.8 montrent le taux de réception cumulé moyen en utilisant la Stratégie d'ordonnancement Aléatoire. Ces résultats sont comparables à ceux obtenus en utilisant la méthode analytique décrite ci-dessus et illustrée dans la figure 5.8 ÉTIQUETTE [A] (en utilisant le taux de perte observé et la même vitesse moyenne à l'entrée du tronçon). Nous notons que cette stratégie est moins efficace, mais beaucoup plus prévisible. D'autre part, elle est plus équitable que la Stratégie Séquentielle. Cependant, en regardant le tableau 5.2, nous pouvons constater que cette technique permet à peine de terminer un téléchargement. Ceci est confirmé par le modèle analytique.

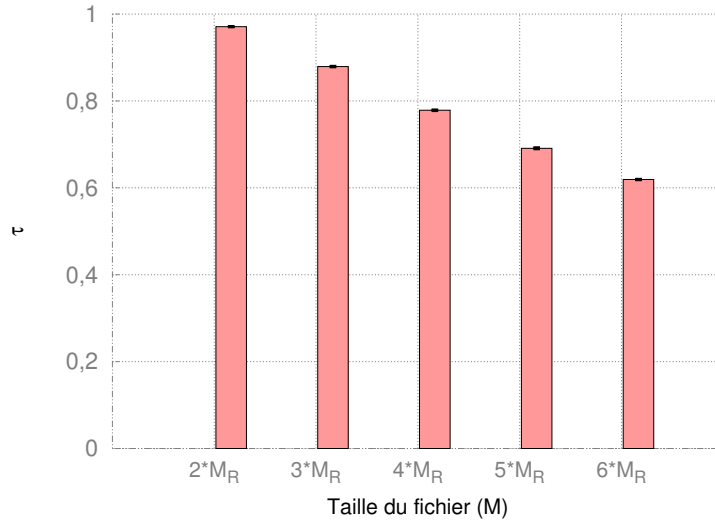


FIGURE 5.7 – τ en fonction de M avec la Stratégie Aléatoire

TABLE 5.2 – Taux de OBU_s avec le fichier complet en utilisant la Stratégie Aléatoire

M	numéro de la RSU						Total
	1	2	3	4	5	6	
1 * M_R	0,03	1,06	43,21	43,58	10,15	1,68	99,71
2 * M_R	0,0	0,0	0,0	0,0	0,0	0,1	0,1

Les deux modèles (simulation et analytique) montrent également qu'avec l'aide des réceptions de doublons, le taux de perte influence peu les résultats. La figure 5.9, par exemple, montre le taux moyen de réception τ (évalué analytiquement) pour $M = 3300$ morceaux de fichier en fonction du nombre de RSU_s pour différents taux de perte de

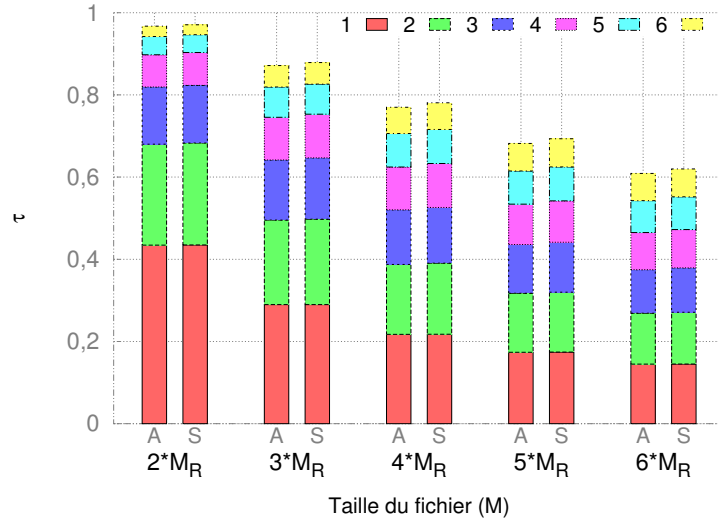


FIGURE 5.8 – τ en fonction de M avec la Stratégie Aléatoire, τ est dégroupé par RSU ([A] Analytique, [S] Simulation)

paquets.

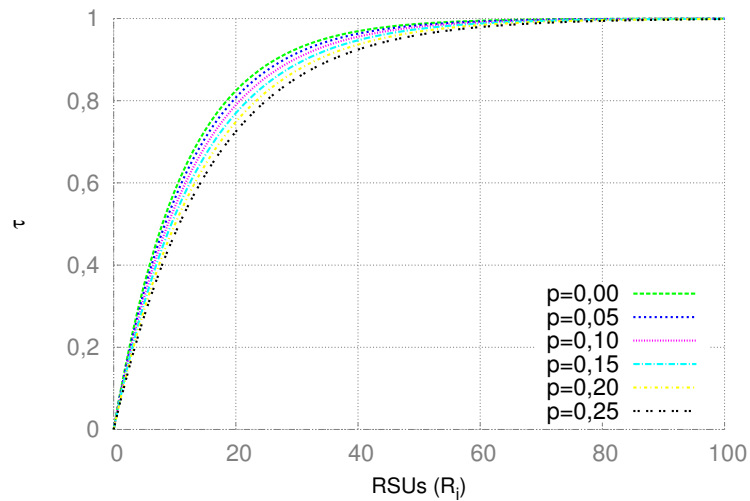


FIGURE 5.9 – Taux de réception τ (évalué analytiquement) en fonction du nombre de RSUs, pour différents taux de pertes de paquets

Le ratio π_1 dépend de la taille de la carte et du nombre de **RSUs**. Ce ratio augmente rapidement en passant à côté des premières **RSUs**, puis n'augmente plus que très

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

doucement.

Nous allons maintenant évaluer si le schéma unicast peut aider à accélérer cet accroissement final.

5.6.2.3 Utilisation conjointe des schémas multicast et unicast

Le schéma unicast est manifestement inefficace, sauf si le système est très peu chargé. D'autre part, avec un système utilisant le schéma multicast pur, l'ordonnancement séquentiel peut provoquer un comportement imprévisible et inéquitable. En revanche, il conduit à une bonne performance du ratio de téléchargement complet. Un ordonnanceur aléatoire conduit à des résultats plus prévisibles et équitables, mais avec un taux plus faible à la fois dans la réception moyenne (τ) et dans les téléchargements terminés (π_1).

Nous avons fait des simulations en partageant le débit (C_a) entre le multicast et l'unicast, à l'aide des paramètres α et le seuil τ_s décrits dans la section 5.2.3.3.

L'idée essentielle est que la transmission en unicast, tout en consommant des ressources qui pourraient être utilisées pour le multicast, aide certains véhicules à terminer leur téléchargement, et donc augmenter l'efficacité globale et l'équité.

Nous avons observé dans nos résultats que quand le scénario utilise un seuil faible, la bande passante disponible pour le multicast est réduite considérablement. Comme le seuil est bas, en fait, la plupart des morceaux manquants n'ont pas encore été envoyés aux véhicules. Il existe également une forte corrélation entre les paquets reçus par les différents véhicules. Cela incite à compléter au travers d'une transmission multicast des paquets manquants au plus grand nombre. Nous y reviendrons dans la sous-section 5.6.2.4. Symétriquement, les valeurs de seuil élevées sont également inutiles, car les véhicules tardent trop à atteindre ce seuil.

Enfin, toute diminution du débit de la transmission en multicast a un impact plus fort sur les résultats que l'augmentation offerte par l'utilisation des transmissions en unicast. La figure 5.10 montre les résultats en utilisant la stratégie séquentielle pour le multicast. La figure 5.11 montre les résultats en utilisant la stratégie aléatoire pour le multicast. Les figures retracent les résultats pour les seuils $\tau_s = 0,5$, $\tau_s = 0,7$ et $\tau_s = 0,9$.

5.6.2.4 Multicast avec la méthode du morceau le plus recherché d'abord

Afin de clarifier la possibilité d'améliorer la performance du multicast, la stratégie qui cherche à envoyer d'abord le morceau le plus recherché a également été mise en œuvre. Dans ce scénario, chaque véhicule transmet son bitmap à chaque fois qu'il

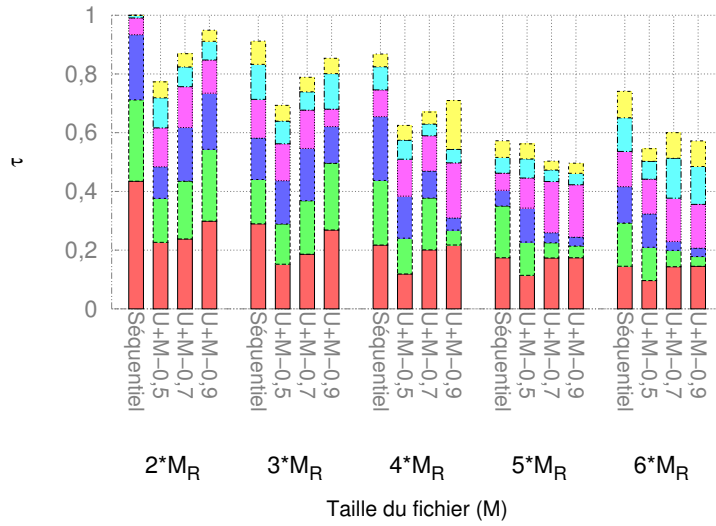


FIGURE 5.10 – τ en fonction de M avec la stratégie unicast plus multicast en utilisant l’algorithme séquentiel

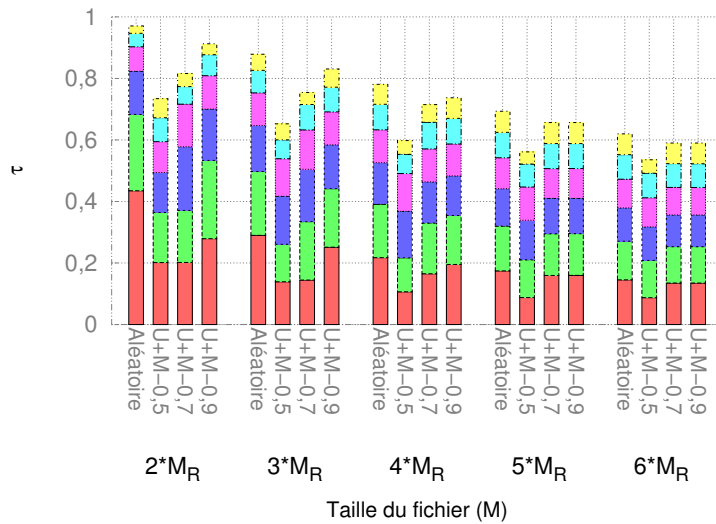


FIGURE 5.11 – τ en fonction de M avec la stratégie unicast plus multicast en utilisant l’algorithme aléatoire

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

rentre dans la zone de couverture d'une [RSU](#).

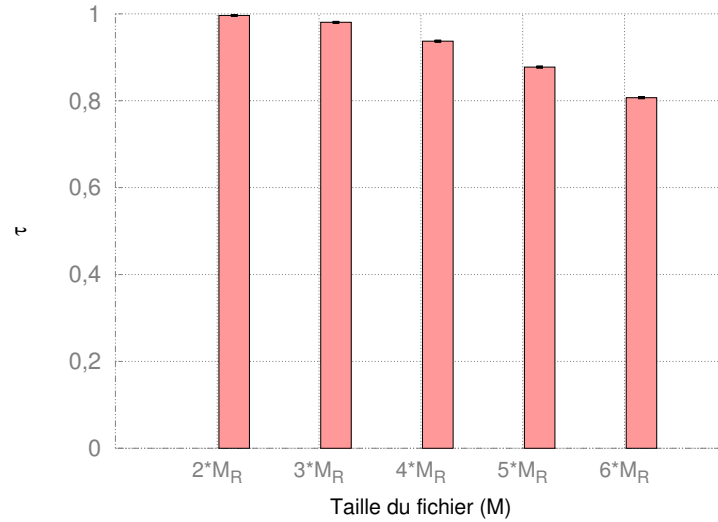


FIGURE 5.12 – τ en fonction de M avec la stratégie du morceau le plus recherché d'abord

Les figures [5.12](#) et [5.13](#) montrent le taux de réception cumulé moyen en utilisant l'ordonnanceur proposé dans cette section. Nos résultats montrent une augmentation significative du ratio moyen des morceaux reçus. Cependant, le problème principal reste toujours le nombre de morceaux doublons reçus par les [OBUs](#).

En conséquence, même si cette stratégie est plus rapide que les stratégies séquentielles et aléatoires pour télécharger une forte proportion du fichier, avoir une probabilité forte de téléchargements complets est encore difficile à atteindre.

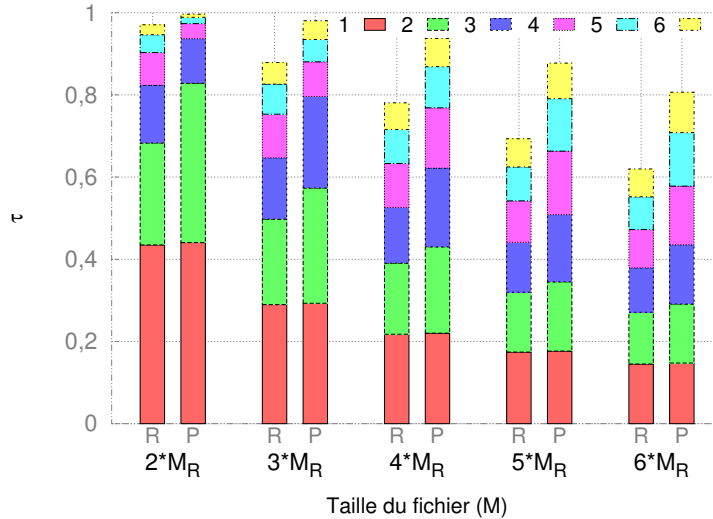


FIGURE 5.13 – τ en fonction de M avec la stratégie du morceau le plus recherché d’abord, τ est dégroupé par RSU ([R] Aléatoire, [P] Le morceau le plus recherché d’abord)

5.7 Conclusions

Nous croyons que l’utilisation de certaines RSUs situées le long de l’autoroute peut être utile pour une application de téléchargement de contenus, en particulier dans les scénarios du trafic peu dense. Dans ce chapitre, nous avons étudié l’utilisation d’une infrastructure de réseau, et comparé des schémas de délivrance multicast, unicast et combinés. Nous avons évalué le taux de réception moyen et le taux des voitures qui ont réussi à télécharger le fichier complet. Nous avons montré que l’utilisation du multicast seul était la meilleure solution.

Nous nous sommes alors focalisés sur la stratégie d’envoi de paquets pour le multicast. Nous avons tout d’abord retenu les stratégies séquentielle et aléatoire. La stratégie aléatoire est prévisible comme nous l’avons constaté en comparant des simulations avec un modèle analytique simple. Les performances sont en revanche décevantes.

Nous avons étudié aussi la stratégie du morceau le plus recherché d’abord qui est fondée sur des informations de feedback. Ses résultats sont prometteurs et améliorent sensiblement les résultats des deux solutions précédentes.

Nous avons ensuite constaté que plusieurs paramètres sont à prendre en compte tels que la densité de RSUs, le débit disponible et la taille du fichier. Les résultats montrent que l’indicateur le plus décisif n’est pas le taux de perte de paquets, mais le taux de

5. PASSAGE À L'ÉCHELLE - COMPARAISON UNICAST MULTICAST

doublons reçus.

L'unicast n'est évidemment pas une solution pour passer à l'échelle. Le multicast peut être configuré de manière à atteindre un ratio élevé de nouveaux paquets reçus, mais il est difficile d'assurer un taux élevé de téléchargements terminés, en particulier en raison, encore une fois, du taux élevé de doublons.

Chapitre 6

Codage Réseau pour le téléchargement des données par infrastructure

6.1 Description du Problème

Dans le chapitre précédent, nous avons mis en évidence que le principal souci des solutions multicast pour télécharger des fichiers à partir de plusieurs **RSUs** consécutives provenait de la réception de trop nombreux doublons. Il nous est apparu nécessaire de travailler, non pas sur la stratégie d'ordonnement des messages pour laquelle nous avons montré les limites, mais sur le codage de l'information.

6.1.1 Le problème du multicast : morceaux de fichier doublons

Dans ce contexte, le schéma de délivrance unicast est hors de propos. Le multicast permet le passage à l'échelle, grâce à l'utilisation de plusieurs **RSUs** réparties le long de la route. Ces **RSUs** envoient périodiquement le fichier comme une séquence de morceaux.

Nous avons démontré à l'aide d'un modèle analytique (confirmé par des simulations) que la performance de ce système est affectée principalement par les doublons. Notons en outre qu'ils ont toutefois un effet bénéfique, car plus un véhicule traverse de **RSUs**, plus la perte de paquets est compensée par ces doublons. En revanche, en raison de ces morceaux doublons, il est difficile de réaliser le téléchargement complet du fichier.

6. CODAGE RÉSEAU POUR LE TÉLÉCHARGEMENT DES DONNÉES PAR INFRASTRUCTURE

6.1.2 Amélioration du taux de réception

Afin de réduire les doublons, deux types de techniques peuvent essentiellement être envisagés. Le premier consiste à faire connaître au serveur les morceaux reçus par chaque voiture, afin qu'il puisse décider les morceaux les plus appropriés à transmettre à travers chaque *RSU*. Le deuxième consiste à réduire les sources de corrélation qui induisent ces morceaux doublons.

6.1.3 Les techniques fondées sur le feedback

Afin d'appliquer ces techniques, les véhicules peuvent envoyer des informations de feedback vers le serveur, de telle sorte que ce dernier puisse choisir le morceau le plus adapté. Cette stratégie est utilisée avec succès dans la dissémination collaborative où les véhicules échangent un bitmap qui décrit les morceaux déjà reçus. Les morceaux peuvent être envoyés en utilisant la stratégie « le plus rare en premier » [14, 38, 44]. Nous avons utilisé cette même technique avec une communication *I2V* dans le chapitre précédent (*MPRP*); elle a conduit à de bons résultats, mais le taux de réception reste insuffisant pour que l'application soit satisfaisante pour les utilisateurs, en particulier, dans le cas où seul un taux de réception de 100% est nécessaire.

6.1.4 Les techniques fondées sur la diminution de sources de corrélation

Ces techniques vont essayer de minimiser la probabilité de recevoir des morceaux doublons sans l'aide d'informations de feedback.

Le phénomène de doublons a été observé en utilisant les différentes stratégies d'ordonnancement. Dans le cas d'une émission séquentielle des morceaux du fichier. Ce phénomène est dû essentiellement à une forme de synchronisation entre le véhicule qui passe par les *RSUs* et les transmissions périodiques à travers ces *RSUs*. Il dépend de la vitesse du véhicule, du débit et de la distance entre les *RSUs*.

Nous avons constaté que le nombre de doublons diminue lorsque ces morceaux sont envoyés à travers chaque *RSU* avec une permutation aléatoire. Cette solution est également plus équitable entre les véhicules, mais n'améliore pas le ratio de téléchargement du système. Elle peut même diminuer légèrement ce ratio global, car le phénomène de synchronisation mentionné précédemment peut aussi conduire à un ratio de téléchargement élevé pour certains véhicules.

Une autre façon de réduire la probabilité de morceaux doublons consiste à augmenter considérablement le nombre de morceaux différents. Si nous sommes toujours en

mesure de construire le fichier avec plus de M morceaux (temporairement sans tenir compte de l'overhead), les performances du système pourraient augmenter.

Le codage réseau linéaire est alors l'outil parfaitement adapté[41, 65]. Avant de décrire la stratégie « [Pseudo Random Network Coding in Vanet for Data Download \(PRAVDA\)](#) », nous précisons notre modèle du système.

6.2 Modèle du Système

Nous allons étudier le système en utilisant à la fois l'évaluation analytique, et la simulation. Évidemment, le modèle utilisé pour l'évaluation est plus simple que celui utilisé pour la simulation. Ci-dessous, nous décrivons ces modèles.

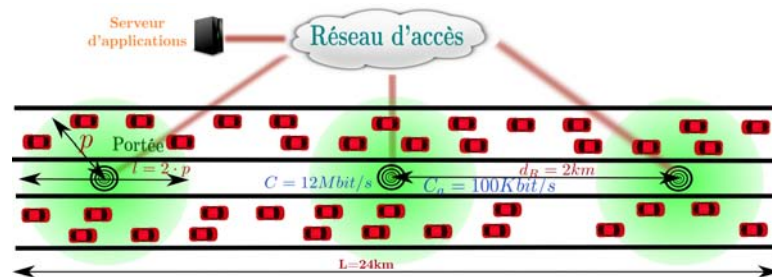


FIGURE 6.1 – Scénario

6.2.1 Modèle de Mobilité

Le schéma général du scénario a été présenté dans le chapitre 4. Le modèle de mobilité exploité dans ce chapitre est l'IDM/MOBIL[62, 63]. Les interarrivées de véhicules suivent une loi exponentielle de taux $\lambda = 0,4\text{ veh/s}$. Tout comme dans le chapitre précédent, la vitesse du véhicule à l'entrée de l'autoroute est donnée par une distribution normale de moyenne $\bar{\mu} = 36\text{ m/s}$ et un écart type de $\sigma_{\mu} = 1\text{ m/s}$. La longueur (L) de l'autoroute est de vingt-quatre kilomètres.

Pour l'évaluation analytique, nous simplifions ce modèle. La vitesse des véhicules est considérée comme constante et IDM/MOBIL n'a pas été pris en compte.

6.2.2 La topologie réseau

Les RSUs sont réparties le long de l'autoroute, écartées d'une distance de deux kilomètres. De cette façon, nous n'avons qu'une couverture partielle de l'autoroute.

6. CODAGE RÉSEAU POUR LE TÉLÉCHARGEMENT DES DONNÉES PAR INFRASTRUCTURE

Nous supposons que chaque véhicule est équipé d'une **OBU**. Ainsi, lorsqu'un véhicule est à l'intérieur de la portée de transmission d'une **RSU**, il peut en recevoir des paquets (en fonction des conditions de propagation).

6.2.3 Le modèle applicatif

Supposons que le fichier soit divisé en M morceaux $\{m_0 \dots m_{M-1}\}$. Le serveur envoie périodiquement une séquence de morceaux à travers chaque **RSU**. Afin d'accroître l'équité, le phénomène de synchronisation, décrit auparavant, est détourné en utilisant un ordre aléatoire pour chaque **RSU**.

Au cours de son périple, chaque **OBU** recevra des morceaux de plusieurs **RSUs** consécutifs. Un morceau bien reçu est dit innovant s'il n'a pas déjà été reçu par l'**OBU**.

6.3 Introduction à PRAVDA

Comme indiqué précédemment, nous voulons construire un grand nombre de morceaux ; le fichier devra être constitué de plus de M morceaux (plus le coût de l'overhead), à la condition qu'ils soient différents. Nous allons pour cela utiliser le « **Codage Réseau Pseudo-Aléatoire (PRNC)** ».

Le **PRNC** a été déjà utilisé pour la dissémination de données dans les **VANETs** [44, 47, 48]. Toutefois, comme déjà indiqué, ces études se concentrent principalement sur le téléchargement collaboratif en utilisant une communication **V2V**.

L'utilisation du **PRNC**, pour le multicast en se servant d'une infrastructure, a été proposée pour un contexte WiMax[66]. L'objectif était de permettre le téléchargement d'un fichier tout en réduisant le taux de retransmission.

Le point faible du **PRNC** est l'overhead (requis pour que le récepteur puisse décoder les morceaux codés) dont la taille est fonction du nombre de morceaux (M), il existe donc un compromis entre les deux. Le but de **PRAVDA** est de trouver ce compromis.

6.3.1 Utilisation du codage réseau

Définissons $\mathbf{c} = (c_0, \dots, c_{M-1})$, $c_i \in \mathbb{F}_z$, un ensemble de coefficients du champ de Galois \mathbb{F}_z où $z = 2^\gamma$ et γ est le nombre de bits de chaque coefficient. En appliquant les techniques classiques du codage réseau [41, 67], nous pouvons alors encoder un morceau e comme une combinaison linéaire des morceaux originaux m_i :

$$e = \sum_{i=0}^{M-1} c_i \cdot m_i \quad (6.1)$$

m_i doit être considéré ici comme un vecteur, dont les coefficients sont dans \mathbb{F}_z (il faut tenir compte de l'overhead et du bourrage potentiel si le nombre de bits n'est pas un multiple de γ). Le nombre de dimensions de ce vecteur est donc $d = \lceil M_b/\gamma \rceil$ où M_b est la longueur en bits du morceau.

Un véhicule est capable de construire le fichier dès qu'il a reçu M morceaux linéairement indépendants. Pour cela, nous avons besoin de connaître le vecteur \mathbf{c} des coefficients associés à chaque morceau.

À cet effet, nous pouvons imaginer que les coefficients soient inclus dans les morceaux encodés. Cette technique est utilisée par exemple dans le protocole proposé par [48]. Cependant, avec de gros fichiers, cela pourrait conduire à un overhead très important. Chaque vecteur de coefficients peut être considéré comme $M \cdot \gamma$ bits, potentiellement des centaines d'octets, en fonction de la taille du fichier.

La figure 6.2 montre le ratio de l'overhead (longueur de l'overhead divisée par la taille du fichier) en fonction de la taille du fichier et de la taille des morceaux. Nous constatons que des grandes valeurs (proche de 50%) peuvent être atteintes facilement.

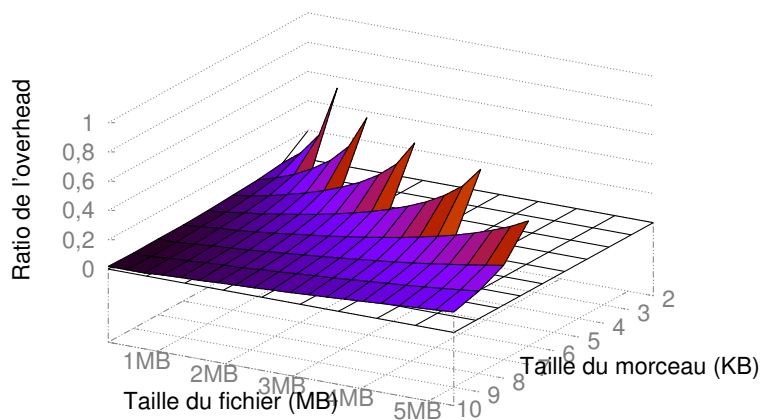


FIGURE 6.2 – Ratio de l'overhead pour les coefficients incorporés

Une autre solution consiste à utiliser une table de codage prédéfinie et inclure, dans chaque morceau codé, l'indice du vecteur de coefficients[66]. Toutefois, il est nécessaire d'avoir un grand nombre de morceaux codés afin d'atteindre une faible proportion de morceaux doublons. Cela pourrait conduire à nouveau à une grande base de données.

Nous avons choisi une troisième voie dans laquelle le récepteur est capable de re-

6. CODAGE RÉSEAU POUR LE TÉLÉCHARGEMENT DES DONNÉES PAR INFRASTRUCTURE

construire les coefficients de lui-même.

6.3.2 Codage réseau pseudo aléatoire

Le PRNC a montré son efficacité dans un contexte légèrement différent[68]. Notre objectif est de construire un grand nombre de morceaux codés afin de réduire la probabilité de morceaux doublons. Pour ce faire, chaque paquet codé est associé à une graine. Un « Générateur de Nombres Pseudo-Aléatoires (PRNG) » est ensuite utilisé avec une graine de départ pour calculer le vecteur correspondant de coefficients \mathbf{c} .

Chaque fragment codé est construit en utilisant la procédure suivante. Le serveur choisit une graine de manière aléatoire $s \in [0 \dots (2^\beta - 1)]$, où β est le nombre de bits de la graine. Supposons que $\beta = 32$. Ensuite, PRNG utilise cette graine pour construire M coefficients c_0, \dots, c_{M-1} , où $c_i \in \mathbb{F}_z$. Le morceau codé e_s est calculé en utilisant la formule (6.1).

Le morceau et la graine associée sont envoyés dans un seul paquet. En conséquence, l'overhead est très faible, seulement quelques octets ($\beta/8$) pour décrire la graine. À titre de comparaison, sans codage réseau, le numéro du morceau i doit être envoyé avec le morceau non codé m_i , ce qui conduit au même overhead. Si la taille du paquet est de l'ordre du millier d'octets, cet overhead est de moins de 1% de M . Le serveur envoie à travers chaque RSU une séquence aléatoire de ces morceaux codés.

Le principal inconvénient est que pour décoder le fichier, le récepteur requiert M morceaux différents, linéairement indépendants. Puisque les coefficients sont générés de façon (pseudo) aléatoire, il existe une probabilité non nulle que l'ensemble de morceaux M n'ait pas cette propriété. Par conséquent, un récepteur peut avoir besoin de $M + \epsilon$ morceaux différents pour être en mesure de décoder le fichier. L'objectif de la section suivante est d'évaluer ϵ .

6.4 Modèle analytique

Le taux de perte utilisé dans ce modèle a été déterminé au moyen de simulations. En utilisant les paramètres décrits dans la section 6.5, nous avons mesuré un taux de perte de paquets moyen de $p_e = 17\%$.

Un véhicule reçoit plusieurs morceaux au cours de son voyage. Nous évaluons la probabilité que le véhicule puisse décoder un ensemble de morceaux suffisant pour obtenir le fichier.

6.4.1 Nombre de morceaux de fichier reçus

Si à l'instant t , k morceaux ont été envoyés vers un véhicule (à travers les différentes **RSUs** au long de la route), il a reçu correctement en moyenne $\bar{r} = (1 - p_e) \cdot k$ morceaux, où $1 - p_e$ est la probabilité de recevoir un paquet sans erreur. Pour simplifier, supposons que p_e soit constant, bien que p_e soit fonction de la distance entre l'**OBU** et la **RSU**.

6.4.2 Nombre de graines différentes

Chaque morceau est construit sur la base d'une graine générée de façon aléatoire. Par conséquent, si l'on note r le nombre de morceaux reçus correctement, il faut déterminer la probabilité $\Psi_r(d)$ que d de ces morceaux aient été engendrés avec une graine différente (et qui sont donc différents).

Pour cela, il convient de noter que pour avoir d morceaux différents après la réception du r -ème morceau, le véhicule doit avoir d ou $d - 1$ morceaux différents avant cette réception (avec $S = 2^\beta$ graines potentielles).

$$\begin{cases} \Psi_r(d) &= \Psi_{r-1}(d) \cdot \frac{d}{S} + \Psi_{r-1}(d-1) \cdot \frac{S-(d-1)}{S} \\ \Psi_r(d) &= 0 \quad \text{si } d > r \\ \Psi_0(0) &= 1 \end{cases} \quad (6.2)$$

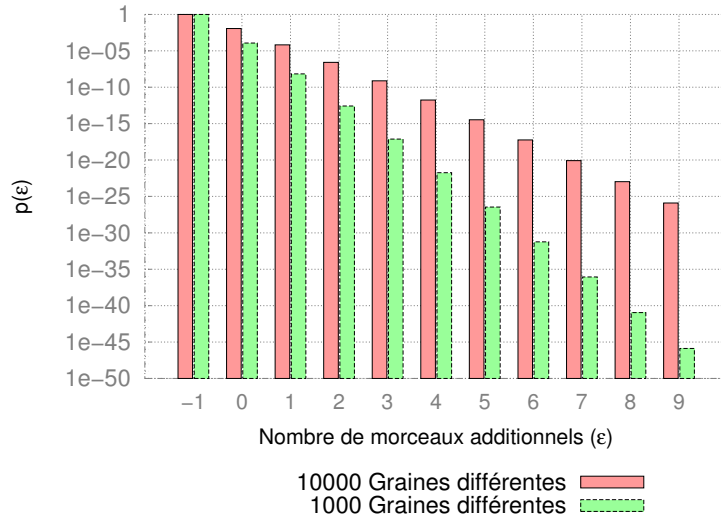


FIGURE 6.3 – Probabilité d'avoir M graines différentes quand $M + \epsilon$ morceaux ont été reçus

6. CODAGE RÉSEAU POUR LE TÉLÉCHARGEMENT DES DONNÉES PAR INFRASTRUCTURE

Ainsi, la figure 6.3 montre la probabilité $p(\varepsilon)$ (en échelle logarithmique) qu'un récepteur ait au moins M morceaux avec des graines différentes après avoir reçu $M + \varepsilon$ morceaux correctement. Dans cette figure, $\beta = 32$, et les résultats sont donnés pour $M \in \{1000, 10000\}$ et $\varepsilon \in [-1 \dots 9]$.

Il est donc clair que, même pour les fichiers de grande taille (M grand), le récepteur n'a besoin que de très peu de morceaux supplémentaires pour les décoder avec une très forte probabilité. Une graine de deux octets fonctionne aussi très bien pour les petits fichiers.

À titre de comparaison, la figure 6.4 montre le nombre moyen de morceaux différents reçus après la réception de 1 à 3000 morceaux, en supposant une taille de fichier de 1000 morceaux. En utilisant le codage réseau (avec $\beta = 32$), le nombre moyen de morceaux nécessaires pour décoder le fichier est de 1001. Sans codage réseau, seulement 950 morceaux sont différents après avoir reçu 3000 morceaux.

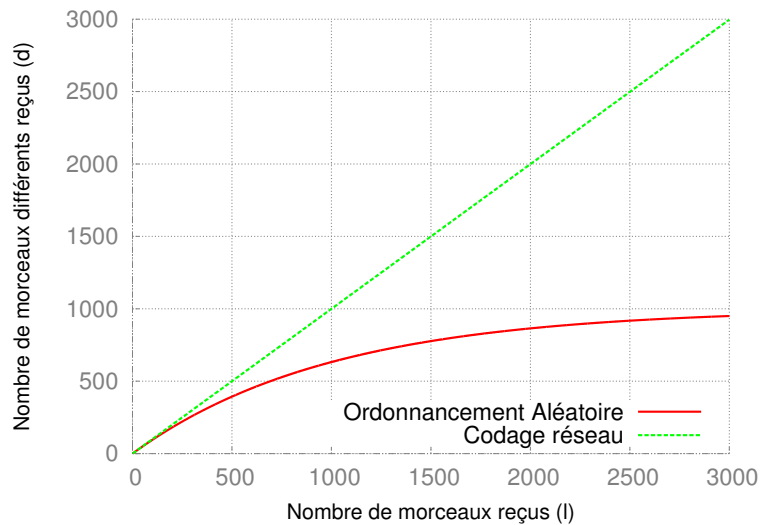


FIGURE 6.4 – Nombre de morceaux différents reçus

Bien entendu, ces résultats ne peuvent pas être pris tels quels. L'inconvénient le plus important de l'utilisation du codage réseau est que le récepteur a besoin de M morceaux pour être en mesure de décoder le fichier. Un autre problème est que ces morceaux ne doivent pas être seulement différents, mais aussi linéairement indépendants.

6.4.3 Nombre de morceaux de fichier linéairement indépendants

Un véhicule ne sera donc pas en mesure de reconstruire le fichier s'il possède au moins M morceaux linéairement indépendants. Par conséquent, si un véhicule a reçu $M + \varepsilon$ morceaux différents, on a besoin de savoir combien de ces morceaux sont indépendants. Pour ce faire, nous allons définir la probabilité $\Phi_r(l)$ que l morceaux soient linéairement indépendants dans l'ensemble des r morceaux reçus et différents.

Pour déterminer $\Phi_r(l)$, nous devons nous rappeler que chaque morceau est un vecteur à M dimensions, dont les coefficients sont dans un ensemble fini de cardinal $z = 2^\gamma$. Ainsi, un ensemble de k vecteurs (ou morceaux) linéairement indépendants définit un espace vectoriel de dimension k de cardinal z^k . La probabilité que le vecteur suivant soit linéairement indépendant de cet ensemble est :

$$\pi_k = 1 - \frac{z^k}{z^M} \quad (6.3)$$

En utilisant la même approche que pour $\Psi_r(d)$, nous trouvons

$$\begin{cases} \Phi_r(l) &= \Phi_{r-1}(l) \cdot (1 - \pi_l) + \Phi_{r-1}(l-1) \cdot \pi_{l-1} \\ \Phi_1(0) &= \frac{1}{z^M} \\ \Phi_1(1) &= 1 - \frac{1}{z^M} \end{cases} \quad (6.4)$$

$\Phi_1(0)$ correspond à la probabilité que le vecteur choisi soit le vecteur nul ; $\Phi_1(1)$ est son complémentaire à 1. La figure 6.5 montre la probabilité d'avoir moins de M morceaux linéairement indépendants après la réception d'au moins M morceaux différents ($\gamma = 8$). Encore une fois, nous avons pris une échelle logarithmique.

Des valeurs différentes de M conduisent aux mêmes résultats. Dans la mesure où z est grand, la situation la plus probable avec moins de M morceaux indépendants est d'avoir $M-1$ morceaux indépendants. Sa probabilité est donc donnée par $\Phi_M(M-1) \approx \Phi_{M-1}(M-1) \cdot \frac{1}{z} \approx \frac{1}{2^8} \approx 3,9 \cdot 10^{-3}$ et ne dépend pas de M .

6. CODAGE RÉSEAU POUR LE TÉLÉCHARGEMENT DES DONNÉES PAR INFRASTRUCTURE

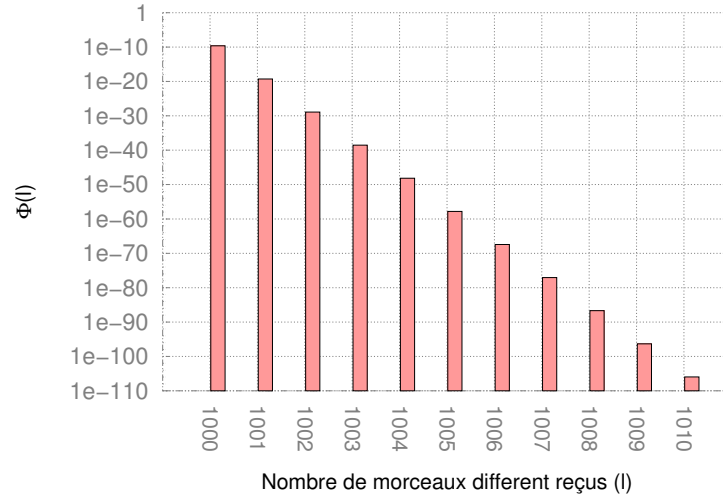


FIGURE 6.5 – Probabilité d’être incapable de reconstruire le fichier en fonction du nombre de morceaux reçus correctement

6.5 Paramètres de simulation

Dans ce chapitre nous avons utilisé la version 15 de NS3. Cependant, nous avons repris tous les paramètres du chapitre précédent. Il est à noter que nous utilisons un débit en couche applicative de $C_a = 100Kbit/s$ comme nous l’avons fait dans le chapitre précédent.

La version standard du protocole 802.11p n’est probablement pas la meilleure pour les applications qui utilisent le multicast. Afin d’améliorer ces applications qui utilisent une communication unidirectionnelle d’infrastructure à véhicules, un canal dédié pourrait être réservé à ce type de communication.

6.6 Résultats de simulation

Afin de comparer les performances de [PRAVDA](#) avec une technique classique de multicast, nous les avons mis en place toutes les deux dans notre environnement de simulation. Nous avons également utilisé le modèle analytique pour évaluer sa pertinence.

La figure [6.6](#) montre le taux de réception moyen pour les véhicules circulant le long de la section de route décrite dans la figure [6.1](#). Dans la figure, M est fonction de M_R . M_R est choisi de sorte qu’un véhicule roulant à la vitesse moyenne (sans embouteillage)

pourrait recevoir M_R morceaux à travers une seule **RSU** (en supposant un taux de perte nul). La taille du fichier varie de $4 \cdot M_R$ jusqu'à $12 \cdot M_R$.

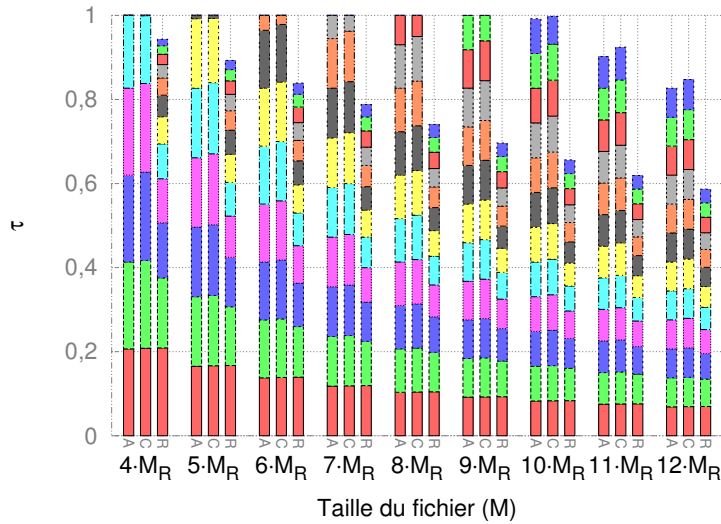


FIGURE 6.6 – Taux de réception moyen pour les différentes techniques en fonction de la taille du fichier (M), étiquettes [A] Analytique, [C] Codage Réseau, [R] Aléatoire

Trois techniques sont évaluées dans cette figure

- Le modèle analytique de **PRAVDA** (étiqueté A)
- La simulation de **PRAVDA** (C)
- La simulation de la technique de multicast avec ordonnancement aléatoire (R), présentée dans le chapitre 5.

Chaque ratio est représenté comme la somme des ratios reçus de chaque **RSU**. Ces valeurs sont soit calculées pour le modèle analytique, soit estimées par simulation en faisant des moyennes sur un grand nombre de véhicules.

Il faut noter d'abord que les résultats théoriques et expérimentaux de **PRAVDA** sont très proches. Nous pouvons conclure que notre modèle analytique convient bien à notre scénario et donc que les résultats précédents sont fiables.

Nous pouvons également déduire de ce résultat que le phénomène le plus significatif pour **PRAVDA** est la perte de paquets. Les morceaux innovants sont également répartis entre les **RSUs** de sorte qu'un fichier avec des morceaux de taille $k \cdot M_R$ est reçu en moyenne après $((k \cdot M_R)/(1 - p_e))$ **RSUs** (p_e est la probabilité de perte de paquets par erreur de transmission).

6. CODAGE RÉSEAU POUR LE TÉLÉCHARGEMENT DES DONNÉES PAR INFRASTRUCTURE

6.7 Conclusions

Dans ce chapitre, nous avons montré comment la technique de codage pouvait éliminer les phénomènes de doublons que nous avons obtenus avec le multicast. Pour être rentable, une telle solution devrait utiliser le moins de débit possible sur chaque [RSU](#) et gêner le moins possible les autres communications. Nous proposons donc [PRAVDA](#), une solution fondée sur le multicast au travers d'une infrastructure.

Nous avons utilisé le Codage réseau pseudo-aléatoire pour engendrer un grand nombre de morceaux différents. Afin de minimiser l'overhead habituellement nécessaire pour transporter les coefficients, nous avons proposé une nouvelle manière de retrouver ces coefficients au niveau du récepteur.

Nous montrons à la fois avec la simulation et l'évaluation analytique que [PRAVDA](#) peut atteindre de très bonnes performances. En outre, l'utilisation du [PRNC](#) augmente considérablement la diversité des morceaux reçus par les différentes voitures.

Pour compléter notre travail, nous allons maintenant étudier la possibilité d'utiliser en plus des communications [I2V](#), des communications [V2V](#). Ces dernières tireront parti de la grande diversité des messages reçus avec [PRAVDA](#) et permettront de compléter la couverture des [RSU](#).

Chapitre 7

Utilisation des communications V2V pour le téléchargement de fichiers

7.1 Description du problème

Dans les chapitres précédents, nous avons analysé différentes techniques pour envoyer un fichier à travers plusieurs **RSUs** en mettant en évidence la supériorité du multicast sur l'unicast ainsi que l'importance des phénomènes de doublons. Un des objectifs est donc de minimiser ces doublons, dont nous avons vu qu'ils représentaient un facteur limitant des performances de l'application. Nous étudions dans ce chapitre l'utilisation du **V2V** en appui des communications **I2V** afin de finir de recevoir les fichiers et d'améliorer le délai.

Le meilleur gage d'efficacité consiste à pouvoir télécharger directement de la **RSU** en utilisant une communication **I2V**, comme nous l'avons analysé dans les chapitres 5 et 6. Pourtant en raison des restrictions liées au coût élevé de déploiement, les **RSUs**, surtout au début, vont être très dispersées et provoquer des temps de connexion à l'infrastructure courts. Les **OBUs** auront des communications intermittentes avec les **RSUs** (comme l'illustre la figure 7.1), ce qui réduit l'efficacité de l'application. Cependant, il faut prendre en compte que cette intermittence n'est pas liée qu'aux communications **I2V**, mais se retrouvera aussi pour les communications **V2V** si la densité de véhicules par kilomètre est faible.

Le problème d'intermittence a surtout un impact sur le téléchargement des fichiers de grande taille. Si les **OBUs** ne réussissent pas à recevoir tout le fichier depuis la

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

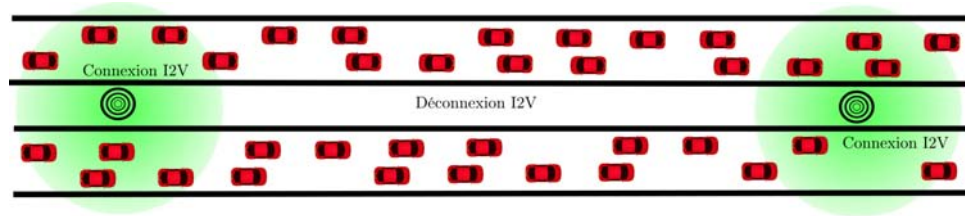


FIGURE 7.1 – Le scénario d’intermittence

RSU courante, elles doivent attendre la prochaine **RSU**. Cependant, les **OBUs** peuvent également profiter de communications **V2V** pour échanger des morceaux manquants. Dans la figure 7.2, nous montrons que si les quatre **OBUs** échangeaient leurs morceaux, au bout de quelques échanges, toutes les **OBUs** obtiendraient le fichier complet.

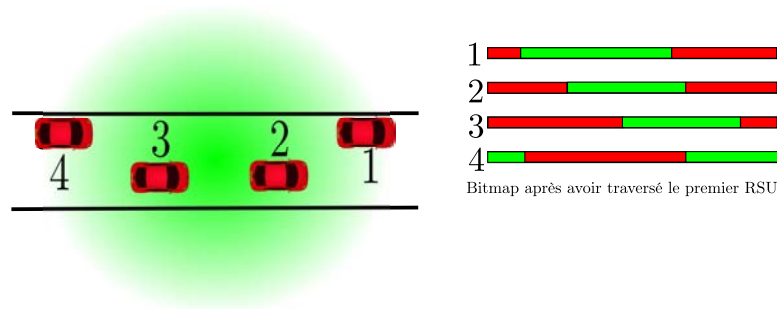


FIGURE 7.2 – Le téléchargement d’un fichier

Pour échanger les morceaux de fichier, les **OBUs** peuvent utiliser différentes stratégies. Nous allons proposer et comparer ces stratégies, en nous fondant sur la littérature abondante en ce domaine, puis détailler leurs avantages et leurs inconvénients dans un scénario avec des voies de circulation à double sens puis à sens unique. Nous voulons tester l’efficacité des communications **V2V** en complément de communications **I2V**.

7.2 Modèle du Système

7.2.1 Modèle de Mobilité

Nous nous fondons ici encore sur le modèle décrit dans le chapitre 4. La longueur de l’autoroute, que nous allons analyser dans ce chapitre, est de vingt-quatre kilomètres. Le modèle de mobilité utilisé est l’IDM/MOBIL[62, 63]. Le taux d’entrée de véhicules

est fixé à $\lambda = 0,4 veh/s$. La vitesse d'un véhicule à l'entrée de l'autoroute suit une distribution normale de moyenne $\bar{\mu} = 36 m/s$ et un écart-type de $\sigma_{\mu} = 1 m/s$.

7.2.2 La topologie réseau

Le scénario général d'une telle application est décrit par la figure 7.3. Les zones couvertes par une **RSU** alternent avec des zones « blanches » dans lesquelles des communications **V2V** sont utilisées. La taille de ces zones blanches est variable en fonction du déploiement des **RSUs**.

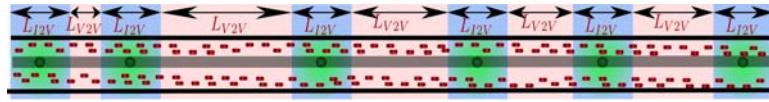


FIGURE 7.3 – V2V en complément d'I2V

Le scénario analysé ici est celui de la figure 7.4. Cette topologie dispose d'une partie couverte par trois **RSUs** sur chaque extrémité du tronçon de dix-huit kilomètres. Nous avons donc douze kilomètres avec des communications **I2V**, six kilomètres à chaque extrémité avec un espacement entre **RSUs** fixe de deux kilomètres. Nous ne considérons pas de communications **V2V** dans les zones couvertes par les **RSUs**. La portion, représentée par la distance L dans la figure, est aussi de six kilomètres, c'est là que va se dérouler la communication **V2V**. En fait, sur l'ensemble de la topologie présentée, nous allons nous focaliser plus particulièrement sur cette portion centrale. Si les **OBUs** n'ont pas réussi à télécharger le fichier au travers des **RSUs**, elles demanderont à leurs proches d'échanger les morceaux manquants.

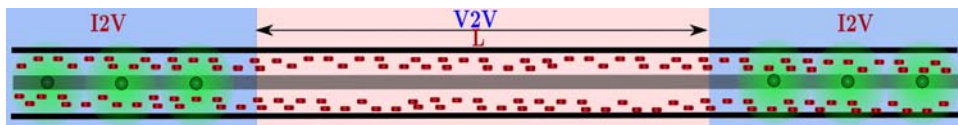


FIGURE 7.4 – Scénario simplifié avec une seule zone blanche

L'idéal serait d'étudier l'ensemble de la topologie. Cependant, la simulation complète est trop coûteuse en mémoire et temps de calcul, si bien que nous procéderons en deux temps. Au cours d'une première campagne de simulations n'utilisant que les communications **I2V**, nous collectons des traces de ce dont disposent les **OBUs** en sortie d'un tronçon d'autoroute (dans la section 7.4, nous analyserons la corrélation entre les

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

morceaux reçus par des véhicules voisins). Dans une deuxième campagne, les **OBU**s sont injectées dans la zone blanche en utilisant ces traces dans le même ordre.

Nous avons ainsi simplifié le processus d'obtention de données depuis les **RSU**s pour les simulations. Les **OBU**s vont entrer avec une partie du fichier des deux côtés de la zone blanche. Nous allons utiliser les traces obtenues dans les chapitres 5 (avec des Stratégies d'ordonnancement) et 6 (avec **Network Coding (NC)**) et essayer de compléter les morceaux déjà reçus par les **OBU**s.

Chaque trace appartenant à une **OBU** contient une séquence d'identifiants de morceaux $T = \{t_0, \dots, t_{\chi-1}\}$ où χ est le nombre de morceaux différents reçus par une **OBU**. Pour les traces venant de la simulation sans **NC**, le nombre d'identifiants possibles est égal au nombre de morceaux du fichier. Pour les traces prises de la simulation avec **NC**, t_i est compris entre 0 et $2^\beta - 1$ où $\beta = 32$ représente le nombre de bits utilisé pour engendrer la graine (voir 6.3.1). Pour le **NC**, on ne conserve que les morceaux linéairement indépendants.

7.3 Description de l'application

Le paradigme utilisé avec la communication **I2V** est le « client - serveur » qui est centralisé. Maintenant, en utilisant la communication **V2V**, nous sommes obligés d'utiliser un paradigme décentralisé, le **P2P**. Ce paradigme permet à chaque véhicule de se comporter à la fois comme serveur (*émetteur*) et client (*récepteur*). Quand un véhicule possède un fragment dont un autre nœud a besoin, il peut lui envoyer.

L'application que nous proposons est fondée sur la technique « push ». Nous avons fait ce choix, car les contenus sont potentiellement destinés à tous les véhicules. Ces derniers pouvant être nombreux, l'utilisation de messages de requêtes de demande de mise à jour de la part des véhicules compromettrait le passage à l'échelle. Nous avons utilisé cette technique pour la communication **I2V** où le serveur envoie le fichier quand une nouvelle mise à jour est produite ; l'**OBU** n'a pas à le demander. Nous allons utiliser la même technique pour la communication **V2V**. Il nous amène à une conception de type dissémination de données. Tout d'abord, nous étudierons les résultats quand un pair *émetteur* utilise l'unicast, comme le fait le **P2P**. Le pair *émetteur* envoie des morceaux sans que le pair *récepteur* le demande.

Le paradigme **P2P** doit exécuter différentes tâches comme la découverte et la sélection de contenus, et également la sélection de pairs. Dans le contexte général, la découverte et la sélection de contenus (ou stratégie d'envoi des morceaux) font référence à plusieurs fichiers, et à leurs morceaux. Dans notre contexte, seul le deuxième

concept a un sens, puisqu'on ne considère que le dernier fichier de mise à jour.

Le contexte sans fil ajoute la propriété de diffusion du support qui permet à peu de frais de faire du broadcast. De plus, le téléchargement depuis les **RSUs** peut avoir été fait avec du **NC** ou non.

Nous avons conçu l'application **P2P** de la façon la plus simple possible. Elle utilise le protocole **UDP**. Nous supposons en particulier que certaines informations élémentaires (position, densité, ...) sont fournies par les applications fonctionnant sur le canal de sécurité. L'application va surtout utiliser la connaissance de l'**OBU** de son voisinage.

Pour améliorer la performance des protocoles **P2P** dans le contexte véhiculaire, les différentes solutions proposées se servent de la proximité physique des nœuds et de la propriété de diffusion du support sans fil. Si un fichier est suffisamment demandé, il y a une forte probabilité qu'un voisin proche le possède.

Pour la communication **V2V**, l'application utilisera, si besoin, toute la bande passante disponible. Celle-ci est fonction de la densité de véhicules, car le support est partagé.

Dans la mise en œuvre d'application **P2P**, les critères discriminants sont la sélection des pairs, la stratégie de découverte de contenus, et la stratégie d'envoi des morceaux. Par conséquent, nous avons étudié plusieurs configurations que nous pouvons présenter de la façon suivante.

7.3.1 Sans **NC** dans la partie **I2V**

7.3.1.1 Utilisation de l'unicast

Sélection des pairs Pour sélectionner le pair le plus adéquat pour échanger les morceaux, nous avons défini une heuristique. Elle consiste à choisir le véhicule le plus proche qui circule en sens opposé, dans le cas du double sens, et le plus proche, dans le cas du sens unique. En utilisant cette heuristique, nous nous assurons une diversité de morceaux plus importante, puisque si nous prenons le plus proche dans le même sens il aura les mêmes morceaux au bout d'une courte période.

Stratégie de découverte de contenus Nous allons utiliser la même stratégie que celle appliquée lors des premiers travaux de communication **P2P** [37]. Ce mécanisme est connu sous le nom de *mécanisme de bavardage* (gossiping). Une **OBU** envoie son bitmap à intervalles réguliers Δ_t pour que les **OBU**s voisines connaissent ses morceaux. Les **OBU**s qui vont dans le même sens n'ont pas besoin de recevoir fréquemment le bitmap, car elles seront dans la couverture les unes des autres pendant longtemps. En revanche, pour celles qui vont dans le sens opposé, les

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

bitmaps devraient être envoyés de façon plus fréquente, car les interactions, entre **OBUs** en sens opposé, seront plus courtes, mais a priori plus fructueuses. Le problème du mécanisme de bavardage est certainement la tempête de diffusion, dont l'impact croît avec la densité. Ce sujet est abordé par [37].

Stratégie d'envoi des morceaux Dans ce chapitre, nous nous sommes intéressés à l'étude de la stratégie d'envoi de morceaux. Dans les travaux réalisés sur les protocoles **P2P**[37, 47, 48], les auteurs proposent des améliorations en tenant compte de la topologie physique du réseau et de cette façon utilisent le pair *émetteur* le plus proche en nombre de sauts. Des travaux prennent en considération la popularité du fichier[26]. Dans tous ces cas, le pair *récepteur* demande les fichiers et les morceaux de fichiers qu'il veut télécharger en unicast.

Stratégie Séquentielle Cette stratégie envoie les morceaux en parcourant la liste O des morceaux que le véhicule possède et que son voisin le plus proche n'a pas de façon séquentielle.

Stratégie Aléatoire Cette stratégie choisit des éléments de O de manière aléatoire, les envoient et les effacent de la liste.

7.3.1.2 Utilisation du multicast

Sélection de pairs dans ce contexte, la sélection des pairs n'a plus d'intérêt, on envoie les messages à tous les véhicules à portée.

Stratégie de découverte des contenus on utilise le mécanisme de bavardage comme pour l'unicast.

Stratégie d'envoi des morceaux dans ce cas, les stratégies aléatoire et séquentielle sont les mêmes que pour l'unicast, on ne se fonde que sur son plus proche voisin. L'idée est d'abord de bien servir celui-ci et par un effet d'aubaine d'en faire profiter les autres. On ajoute une troisième méthode pour laquelle on va gérer la liste des morceaux les plus recherchés (**MPRP**) en se servant de la même technique que dans le chapitre 5. Dans cette technique, une liste est mise à jour chaque fois qu'un nouveau bitmap parvient à un véhicule. Cette liste est construite en se fondant sur tous les bitmaps reçus. La liste est classée en fonction des morceaux les plus recherchés. Chaque morceau est envoyé en la parcourant. Nous nous sommes fondés sur cette stratégie, car il existe beaucoup de littérature en référence à cette méthode et elle nous semble être une bonne option. BitTorrent qui est l'application la plus utilisée pour le **P2P** utilise cette stratégie, connue aussi sous le nom du « morceau le plus rare en premier ».

7.3.2 Avec NC dans la partie I2V

Dans ce cas, l'utilisation du mécanisme de bavardage a un coût rédhibitoire. L'envoi de bitmaps se traduirait par une liste d'identifiants des morceaux. Si l'application veut envoyer cette liste, à la manière d'un bitmap, elle aurait une taille de $2^\beta/8$ octets où $\beta = 32$.

L'absence de mécanisme de bavardage va obliger à mettre en œuvre une stratégie d'envoi de morceaux à l'aveugle. Dans la mesure où l'on envoie des messages à l'aveugle, l'intérêt d'une solution unicast où l'on sélectionne les messages en fonction des données possédées par son voisin disparaît. Par conséquent, on ne retiendra que du broadcast sans la sélection de pairs. Pour finir, l'application sélectionne le morceau à envoyer de façon aléatoire (avec remise). Ce morceau est choisi de la liste de morceaux (T) que l'OBU possède. Les OBUs vont transmettre ce morceau quand elles ont au moins une OBU voisine à leur portée.

Ce raisonnement est accompagné aussi du fait que le NC en I2V a une diversité plus importante que ces concurrents (quand il y a deux sens de circulation). Quand deux véhicules en sens opposés se retrouvent, il est fort probable que tous les morceaux soient innovants grâce à la grande quantité des morceaux différents que chaque RSU peut transmettre.

La table 7.1 montre le récapitulatif de l'ensemble des scénarios que nous avons évalué.

Stratégies	Schémas de délivrance		Stratégies utilisée en I2V	
	Unicast	Multicast	Sans NC	Avec NC
Aléatoire	✓	✓	✓	✗
Séquentielle	✓	✓	✓	✗
MPPR	✗	✓	✓	✗
Aveugle	✗	✓	✗	✓

TABLE 7.1 – Les stratégies et leurs paramètres

7.4 Les traces I2V

Le scénario à partir duquel ont été obtenues les traces se compose de trois RSUs, avec une distance entre les RSUs de deux kilomètres. Nous prenons les traces obtenues sans NC et la Stratégie d'ordonnancement Aléatoire, puis avec le NC.

Nous écartons la stratégie séquentielle et du morceau le plus recherché en premier sur

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

la partie I2V, car elles peuvent provoquer des biais. La stratégie séquentielle est très dépendante des paramètres de la topologie, comme nous l'avons vu dans le chapitre 5. Pour la stratégie MPRP depuis les RSUs, le problème est plus compliqué, car la technique consistant à effectuer deux simulations consécutives n'est plus correcte. En effet, le comportement de la portion V2V aura un impact sur l'ordre effectif dans lequel la RSU enverra ses messages, car certains des véhicules sont déjà passés par la partie V2V.

Nous allons tout d'abord regarder les corrélations entre les morceaux possédés par les OBUs à la sortie de la zone de couverture des RSUs. Dans le cas sans NC, chaque OBU a reçu un nombre de morceaux différents (χ). La figure 7.5 montre que ce nombre est bien approché par une loi normale. Le PDF tracé correspond à la loi normale ayant la même moyenne μ_χ et écart-type que les simulations. μ_χ et σ_χ sont fonction de la taille du fichier, comme nous pouvons le voir sur la figure 7.6. On note que l'écart-type est faible.

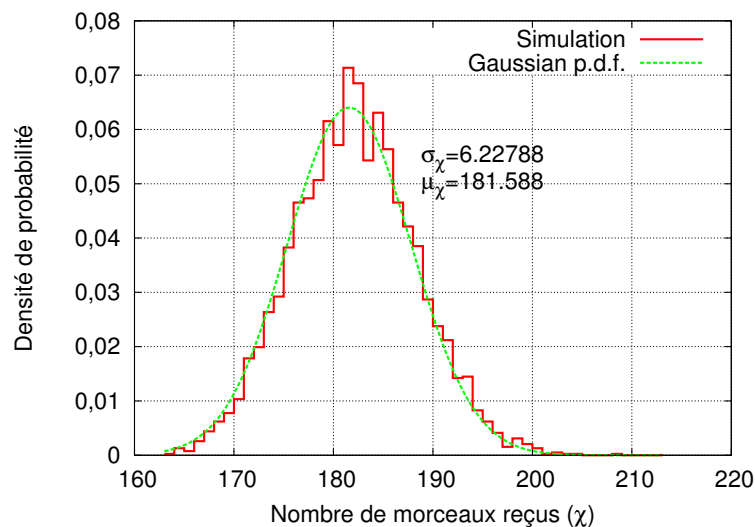


FIGURE 7.5 – Densité de probabilité du nombre de morceaux reçus – I2V sans NC.

Pour mesurer la similarité entre une OBU et ses voisines, nous utilisons la formule (7.1). Ω est estimé depuis les traces. Il est donc calculé à la sortie de la couverture des RSUs.

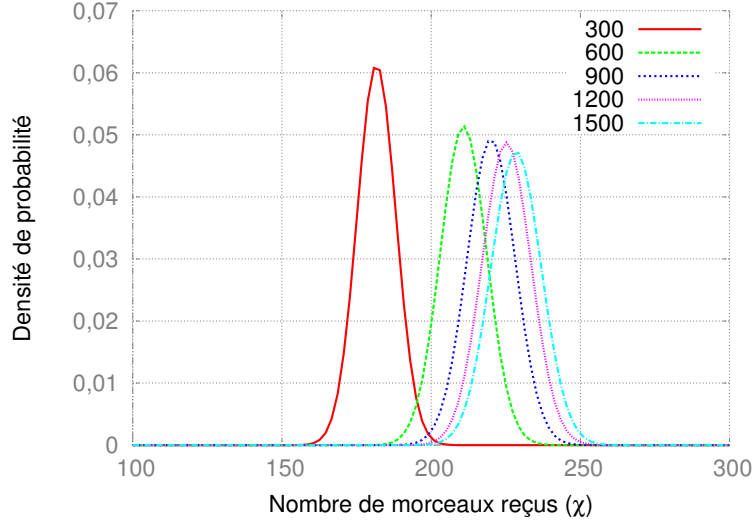


FIGURE 7.6 – Distributions normales en fonction de la taille du fichier – I2V sans NC.

$$\Omega(j) = \sum_{i=1}^N \frac{(|\mathbf{T}^i \cap \mathbf{T}^{i+j}|)}{N} \quad (7.1)$$

où \mathbf{T}^i est l'ensemble des morceaux reçus du véhicule de référence et \mathbf{T}^{i+j} est l'ensemble de morceaux reçus par son $j^{\text{ème}}$ voisin le plus proche en amont ($j > 0$) ou en aval ($j < 0$). La figure 7.7 montre Ω en fonction de $j \in \{-3, -2, -1, 1, 2, 3\}$ et correspond donc aux six OBU les plus proches en sortie de couverture (trois de devant et trois de derrière).

Avec la densité utilisée, une OBU a environ 70% de morceaux communs avec ses deux voisins les plus proches. En outre, Ω décroît avec la taille du fichier. Plus les OBU sont éloignées, moins elles auront de morceaux en commun. Il faut mentionner que ces mesures ont été prises une fois que les véhicules sont sortis de la zone de couverture I2V. D'autre part, les voisins en entrée ne sont pas nécessairement les mêmes en sortie à cause du modèle de mobilité. C'est une des raisons qui explique que la pente ne soit pas très marquée entre j et $j + 1$ dans la figure 7.7.

Regardons maintenant les résultats obtenus dans le cas où le NC est utilisé dans la zone I2V. La figure 7.8 montre que la distribution des morceaux différents reçus est alors indépendante de la taille du fichier. Nous remarquons que la moyenne et la variance du nombre de messages reçus sont fonctions uniquement du nombre de RSUs

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

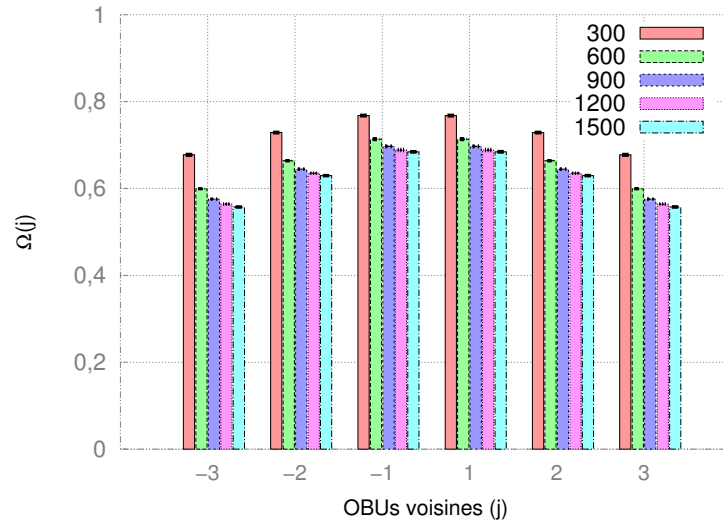


FIGURE 7.7 – Indice de similarité entre OBU voisines les plus proches circulant dans le même sens de circulation

traversées.

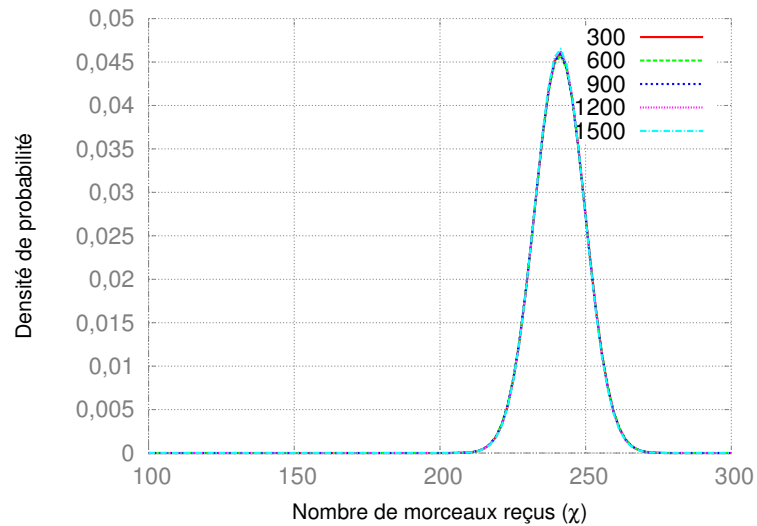


FIGURE 7.8 – La distribution normale en fonction de la taille du fichier avec NC

La figure 7.9 montre que, quelle que soit la taille du fichier, l'indice de similarité est le même. Cet indice est fonction de la proximité entre OBUs voisines.

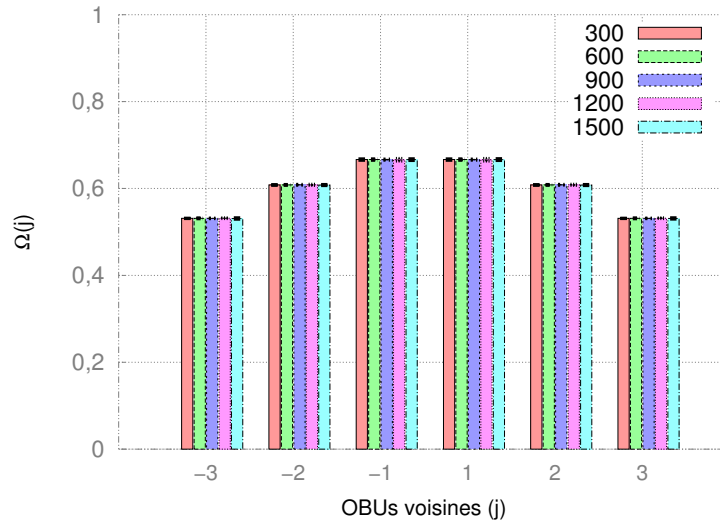


FIGURE 7.9 – Indice de similarité du bitmap entre OBU voisines circulant dans le même sens

7.5 Métriques de performances

Nous avons retenu comme métriques : le taux de délivrance de messages, la latence (le temps nécessaire pour recevoir tout le fichier) et le taux de véhicules ayant terminé la réception du fichier.

7.6 Paramètres de simulation

Le tronçon d'autoroute (L) à analyser est de six kilomètres pour la partie [I2V](#). La configuration utilisée est la même que dans les chapitres précédents. Cependant, nous allons analyser aussi la performance de l'application quand il n'y a qu'un seul sens de l'autoroute, pour mettre en évidence le comportement de l'application dans les deux cas. Nous utilisons un flux véhiculaire de $\lambda = 0,4 \text{ veh/s}$. L'intervalle utilisé pour le mécanisme de bavardage est de $\Lambda = 1 \text{ bitmap/seconde}$. La taille du morceau utilisée est de quatre kilo-octets.

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

7.7 Résultats de simulation

7.7.1 Double sens

7.7.1.1 Les résultats dans le cas où le NC n'est pas utilisé sur la partie I2V

Taux de délivrance La figure 7.10 montre τ en fonction de la taille du fichier M . Avec une taille de fichier de 1500 morceaux, l'ensemble des stratégies, que nous venons de présenter, n'a pas permis à toutes les OBU de télécharger le fichier complet. On était près d'y parvenir puisqu'il manquait moins d'un morceau pour mille en moyenne. Les pires résultats ont été obtenus avec une sélection séquentielle en utilisant l'unicast.

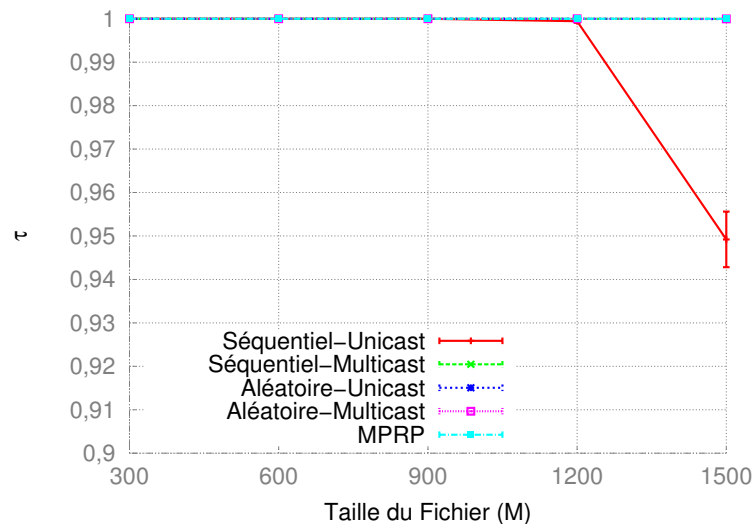


FIGURE 7.10 – τ en fonction de la taille du fichier

Latence En regardant la latence (cf. figure 7.11), nous constatons une croissance exponentielle en fonction de la taille du fichier. La stratégie ayant la plus petite latence est la stratégie aléatoire qui se sert du multicast. Au contraire, la plus grande latence a été obtenue par la stratégie séquentielle et de l'unicast. Pour le fichier le plus grand envisagé, elle prend presque quatre fois plus de temps que la précédente.

Taux de véhicules avec le fichier complet Pour la plupart des applications, un fichier incomplet est inutilisable. Une métrique importante est donc le taux de véhicules ayant le fichier complet. Même si τ avec la stratégie séquentielle et l'unicast reste acceptable, π_1 s'effondre dramatiquement avec des fichiers plus grands (cf.

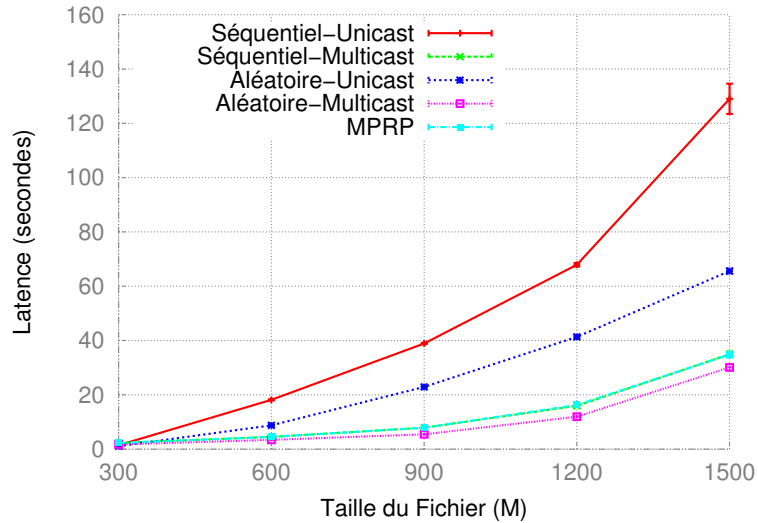


FIGURE 7.11 – Latence en fonction de la taille du fichier

figure 7.12). Avec les autres stratégies, il y a une légère décroissance de π_1 . Quoiqu'il en soit, nous constatons que $M = 1500$ est le point d'inflexion de l'ensemble des stratégies avec ce jeu de paramètres. Ceci est dû au fait que le support devient saturé. Si le tronçon avec V2V est plus grand, l'application peut transmettre un fichier d'une taille plus importante.

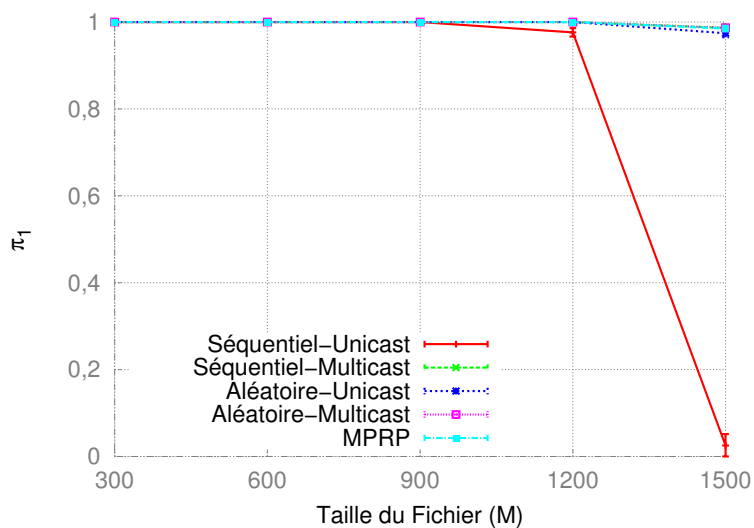


FIGURE 7.12 – π_1 en fonction de la taille du fichier

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

7.7.1.2 Les résultats dans le cas où le NC est utilisé sur la partie I2V

Dans ce cas, en utilisant la stratégie de sélection aveugle aléatoire, τ et π_1 valent 1 : tous les véhicules ont réussi à télécharger complètement le fichier. Dans la figure 7.13, nous avons comparé la latence obtenue dans ce contexte et avec la meilleure stratégie du paragraphe précédent : Aléatoire - multicast mais avec des données envoyées dans la partie I2V sans NC. Le comportement n'est pas très simple à analyser. On peut juste conclure à une supériorité du NC quand la taille du fichier devient très grande.

Il faut faire attention avec de petites tailles de fichier, car un biais est introduit dans les résultats. Ce biais est dû au fait que les stratégies utilisées dans la communication I2V sont différentes. La stratégie utilisée en I2V par la stratégie à l'aveugle est plus efficace que celle utilisée dans la stratégie aléatoire. Le biais est moindre quand la taille du fichier est importante.

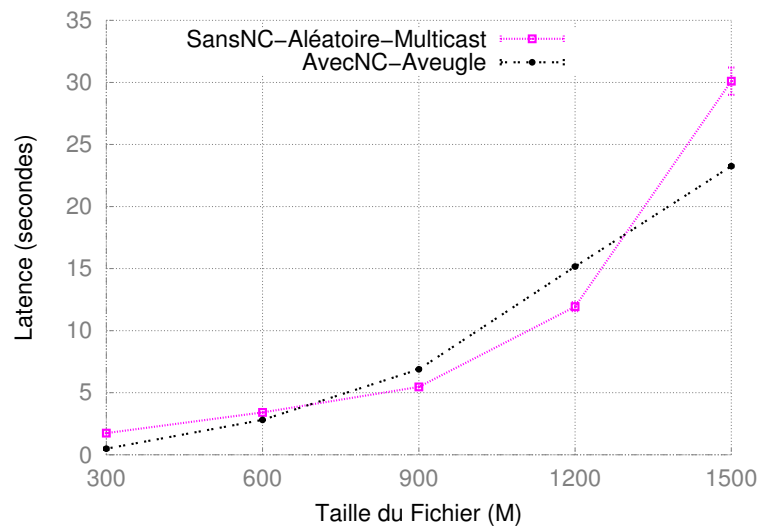


FIGURE 7.13 – Latence en fonction de la taille du fichier – I2V avec ou sans NC

7.7.1.3 Analyse de doublons

Maintenant, nous allons analyser la cause de l'apparition de doublons inattendus. Nous souhaitons pour l'unicast que les OBU fonctionnent par paires. Cependant, en analysant les résultats de la figure 7.14, nous constatons qu'il existe des morceaux doublons, alors que dans les communications unicast, de tels doublons ne devraient pas être présents.

Nous arrivons à la conclusion que notre supposition de départ était fautive. Il peut y avoir plusieurs transmetteurs vers une OBU réceptrice. Notre implantation ne dispose pas de mécanisme pour contrôler ce nombre de transmetteurs, ni de mécanisme pour éviter que ces transmetteurs envoient le même morceau vers un récepteur, c'est à dire qu'un véhicule soit le plus proche voisin de deux véhicules. Ce sont pour ces raisons que de tels doublons apparaissent, et que la stratégie aléatoire obtient de meilleurs résultats que les autres, car avec celle-ci la probabilité de transmettre le même morceau est finalement moins élevée.

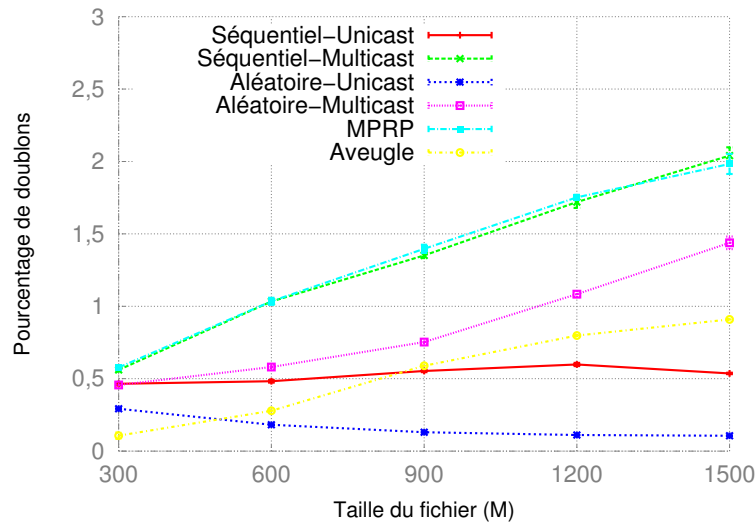


FIGURE 7.14 – Pourcentage de doublons en fonction de M

En utilisant le multicast, la présence de doublons est plus importante à cause de la similarité existante entre OBUs voisines. Si un véhicule a reçu un doublon, il existe une forte probabilité que ce message soit également un doublon pour ses voisins. L'application gère finalement mieux les doublons en se servant de la stratégie aléatoire, tant pour l'unicast que pour le multicast.

Si on regarde la performance de la stratégie à l'aveugle, implantée uniquement avec du multicast, par rapport aux morceaux doublons, nous constatons qu'elle fait mieux que l'unicast quand le fichier est petit, et mieux que le multicast quand le fichier est grand. Avec un fichier de petite taille, elle fait mieux que l'unicast. Cette amélioration considérable est due à la diversité provenant des OBUs circulant en sens inverse, ou bien encore à la quantité plus importante de morceaux reçus par l'I2V.

Pour faire accroître la performance de l'application, il faut donc concevoir un mé-

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

canisme permettant de contrôler l'envoi de morceaux manquants.

Nous pouvons noter sur les figures 7.11 et 7.14 que la latence est directement proportionnelle à la quantité de doublons.

7.7.2 Un seul sens

Dans cette section, nous analysons le comportement de l'application dans le cas où un seul sens de circulation est présent. Nous analyserons cette topologie avec les mêmes métriques que précédemment.

Pour les trois métriques, l'unicast n'a pas eu de bons résultats quelle que soit la stratégie. En utilisant le multicast, l'application a donné de bons résultats même pour des fichiers de grande taille. Les résultats de la stratégie aveugle et le NC se trouvent compris entre ceux de l'unicast et ceux du multicast sans NC.

Dans le cas de l'unicast, τ s'effondre dès que M dépasse 600 morceaux en conduisant à environ 60% pour le fichier le plus grand. La stratégie aléatoire est meilleure que les deux autres même si la marge reste petite. Le multicast a conduit à un τ supérieur à 99% pour les stratégies avec mécanisme de bavardage. La stratégie aveugle a atteint 90% pour la taille du fichier la plus grande (cf. figure 7.15).

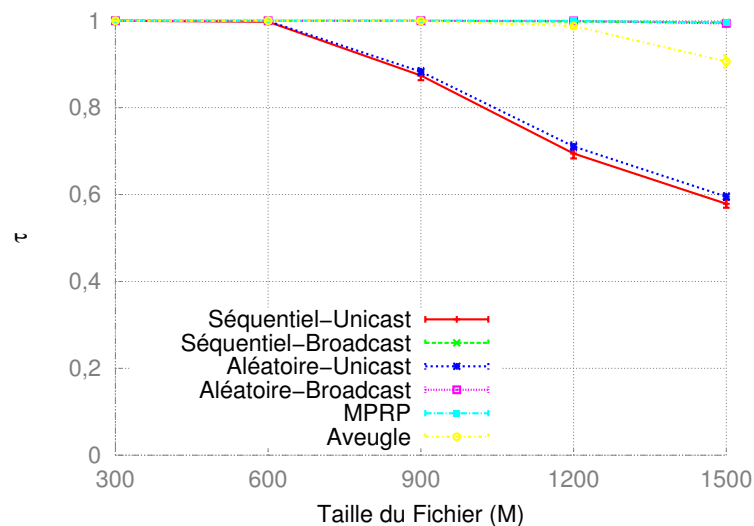


FIGURE 7.15 – τ en fonction de la taille du fichier

La figure 7.16 montre la latence moyenne des véhicules qui ont reçu le fichier complet. L'unicast n'a pas été capable de transférer complètement le fichier quand M dépasse 600 morceaux. En revanche, le multicast a réussi à le faire, en environ 40s pour

la plus grande taille de fichier avec la stratégie aléatoire. Les stratégies avec mécanisme de bavardage ont présenté, dans ce cas, des résultats semblables.

Si l'on s'intéresse au multicast, cf. figure 7.17, les trois stratégies ont des résultats proches. Dans le scénario à double sens de circulation, la stratégie aléatoire a surpassé les deux autres stratégies. Dans le cas du scénario à sens unique, cette stratégie commence à présenter des problèmes pour les fichiers de grande taille. Comme nous pouvons l'apprécier sur la figure 7.16, pour une taille de fichier de 1500 morceaux, la latence de la stratégie aléatoire est plus importante que pour les deux autres. La figure 7.17 nous confirme ce phénomène, puisque la latence est fonction aussi des doublons. La figure 7.17 montre le pourcentage de doublons.

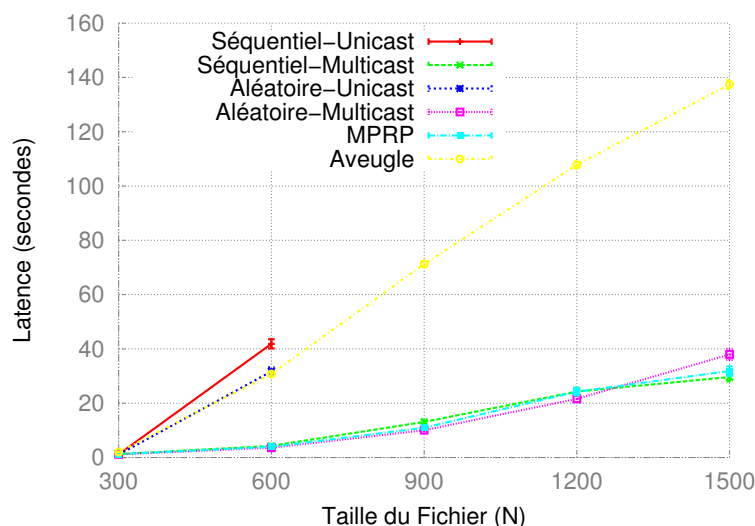


FIGURE 7.16 – Latence en fonction de la taille du fichier

Concernant le taux π_1 de véhicules ayant réussi à télécharger complètement le fichier, nous observons (cf. figure 7.18) qu'avec l'unicast aucune des **OBU**s n'a pu y parvenir, dès que le fichier a une taille M supérieure à 600. Le multicast réussit à transférer le fichier le plus grand pour environ 80% des **OBU**s. La stratégie aveugle et le **NC** a conduit à de bons résultats par rapport à l'unicast, mais pas très bons par rapport au multicast. Avec le fichier le plus grand, la stratégie aveugle a transféré le fichier complet pour environ 30% d'**OBU**s.

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

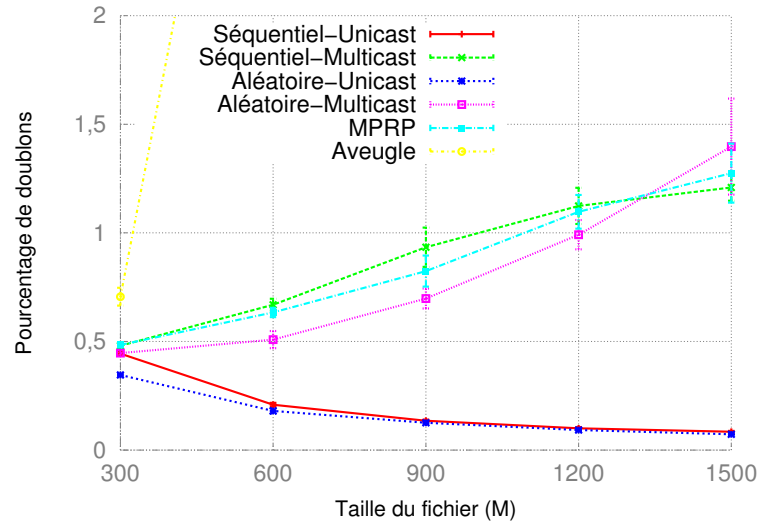


FIGURE 7.17 – Pourcentage de doublons en fonction de M

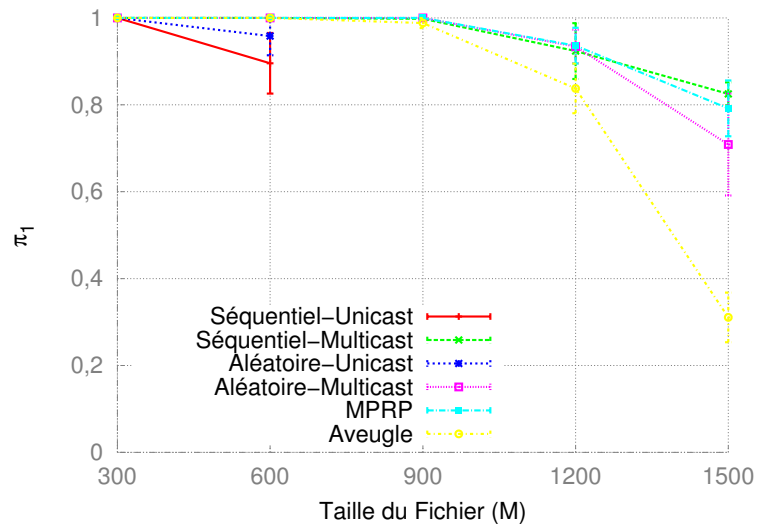


FIGURE 7.18 – π_1 en fonction de la taille du fichier

7.8 Conclusions

Dans ce chapitre, nous nous sommes intéressés à la complémentarité entre communications **V2V** et **I2V**. Les véhicules commencent par recevoir des fichiers en **I2V** et les complètent en **V2V**. Nous avons étudié plusieurs stratégies de communications, de sélection des pairs et des morceaux à émettre.

Quelle que soit la stratégie, nos travaux montrent qu'il vaut mieux utiliser le multicast pour profiter de la propriété diffusante du support et ce d'autant plus que l'application proposée le permet.

Nous avons constaté que quand la corrélation entre les morceaux possédés par les véhicules voisins est forte, la stratégie de sélection du morceau le plus recherché en premier n'améliore pas la performance de l'application.

Quand les communications **I2V** utilisent du **NC**, les performances seront très bonnes quand il y a deux sens de circulation. Cela est notamment dû à la grande diversité produite dans ce cas. Elle trouve ses limites quand il n'y a qu'un seul sens de circulation par rapport à l'absence de **NC** et des stratégies utilisant le mécanisme de bavardage et du multicast. Ce dernier conduit à une meilleure performance, même s'il utilise beaucoup de bande passante.

7. UTILISATION DES COMMUNICATIONS V2V POUR LE TÉLÉCHARGEMENT DE FICHIERS

Conclusions

Les systèmes de transport intelligents constituent un enjeu majeur pour la sécurité routière dans les prochaines années. Leur développement doit se faire en profitant de l'ensemble des moyens de communication qui sont et seront disponibles. Les applications envisagées sont de plusieurs natures en allant de la sécurité routière, à l'aide à la conduite et enfin au confort des utilisateurs qui voudront avoir dans la voiture les mêmes services que dans leur vie quotidienne.

Si les applications de sécurité ont été très largement étudiées en particulier dans le cadre de grands projets internationaux, et c'est bien sûr largement mérité car sans ces applications, le cadre de transport intelligent perd une grande partie de son intérêt, les applications de confort ont été beaucoup moins finement définies.

Nous nous sommes intéressés dans cette thèse à un type d'applications relativement classique qui sera le transfert de fichiers de taille relativement importante mais vers un très grand nombre d'utilisateurs. Ce sera exactement le cas d'une application de téléchargement de cartes routières qui nous a semblé particulièrement emblématique du contexte véhiculaire.

Cette application va se caractériser par un gros volume d'information à transporter vers un nombre de destinataires potentiellement important avec des contraintes faibles de délai.

Dans cette thèse, nous avons essayé de déterminer le ou les moyens de communication les plus appropriés pour mettre en œuvre cette application à un coût réduit pour l'utilisateur.

Le contexte véhiculaire est suffisamment vaste pour couvrir à la fois des réseaux avec et sans infrastructure. Le recours à des solutions avec infrastructure de type réseaux de télécommunications (3G, LTE, satellite) permet de garantir une bonne qualité de service mais à un coût sans doute trop important. Les solutions WiFi actuellement déployées imposent des vitesses de déplacement trop basses et donc ne fonctionnent que sur des aires d'autoroutes ou des parkings ce qui nous a semblé un peu réducteur.

Nous nous sommes donc concentrés dans cette thèse sur les solutions de type [IEEE](#)

7. CONCLUSIONS

802.11p avec des relais (**RSUs**) pour le cas avec infrastructure et le mode ad-hoc du même protocole pour les solutions sans infrastructure.

Si les solutions sans infrastructure (**V2V**) ont donné lieu à une littérature abondante inspirée des réseaux Peer-to-Peer et ad-hoc, les solutions avec infrastructure (**I2V**) n'ont donné lieu qu'à un nombre de travaux très réduit si bien que l'on ne sait pas vraiment comment les véhicules possèdent les informations avant de pouvoir se les échanger. Notre travail s'est donc principalement focalisé sur cette première partie des échanges.

Dans le contexte **I2V**, nous avons imaginé des scénarios autoroutiers où l'on veut diffuser les fichiers vers un grand nombre de véhicules. Notre première contribution a permis de mettre en évidence les limites des solutions unicast qui ne passent pas à l'échelle par rapport à du multicast. Dans le multicast, l'ordre d'envoi des messages s'est révélé un facteur majeur en particulier dans le cas de figure où un utilisateur ne peut pas recevoir tout le fichier depuis une seule **RSU**. Plusieurs stratégies ont été envisagées qui ont montré que si l'on arrive à connaître les morceaux du fichier possédés par les utilisateurs, la meilleure solution sera celle qui consiste à envoyer le morceau le plus manquant en premier, sinon une stratégie séquentielle peut convenir.

Notre résultat le plus marquant a été de mettre en évidence le fait que le principal défaut du multicast était les doublons qui apparaissent au niveau des récepteurs après leur passage devant plusieurs **RSU** et non pas les pertes dues aux erreurs de transmission.

Le recours à de l'unicast pour aider les véhicules qui ont presque tout le fichier à terminer de recevoir les quelques morceaux manquants s'est révélé très décevant car la part du débit que l'on n'utilise pas pour faire du multicast pénalise alors un trop grand nombre d'utilisateurs.

Nous sommes alors arrivés à la conclusion que l'on se devait de travailler non pas uniquement sur l'ordre d'envoi des morceaux du fichier mais sur leur codage. Le recours au Network Coding linéaire aléatoire nous a donc semblé particulièrement approprié pour éviter les phénomènes de doublons. Le serveur envoie ces morceaux codés qui sont alors en très grand nombre, tellement grand que la réception d'un nombre finalement pas déraisonnable de morceaux différents et linéairement indépendants peut se faire rapidement. L'originalité de notre travail a porté sur la façon avec laquelle les coefficients de codage étaient déterminés par le récepteur : plutôt que de les ajouter dans l'en-tête du message, on a préféré lui envoyer la graine du générateur pseudo-aléatoire pour qu'il puisse les recalculer à la réception. Cet algorithme combiné à du broadcast s'est révélé extrêmement efficace.

La dernière partie de notre travail de thèse a porté sur le recours aux communi-

cations **V2V** pour traiter les cas de figure où l'infrastructure des **RSU** est très éparse, ce qui correspondra forcément à la configuration la plus commune dans les premières années de déploiement potentiel des réseaux véhiculaires. On suppose que les utilisateurs ont reçu par l'infrastructure un certain nombre de morceaux et qu'ils essaient alors de s'échanger les messages qui intéressent leurs voisins. Encore une fois, et quelle que soit la façon avec laquelle la solution **I2V** a été implantée, l'unicast ne conduit pas à des résultats intéressants. Dans le cas où l'on n'a pas utilisé le Network Coding sur la partie **I2V**, la sélection des morceaux est importante. Dans le cas du Network Coding, tenir à jour les morceaux qui intéressent ses voisins est rédhibitoire et l'on fait alors une transmission à l'aveugle. Dans tous les cas, pour que les communications **V2V** puissent vraiment aider, il est souhaitable que l'on rencontre des véhicules qui n'ont pas été desservis par les mêmes **RSUs** au même moment car sinon, ils ne nous apporteront rien d'intéressant. Ce phénomène est encore plus flagrant dans le cas de messages reçus avec du Network Coding car on a du mal à s'en rendre compte avec les transmissions à l'aveugle ; dans le cas sans Network Coding sur la partie **I2V**, les véhicules n'échangeront tout bonnement rien.

Perspectives

Cette thèse peut constituer une base à de nombreux travaux futurs qui peuvent s'articuler sur plusieurs axes.

Le premier point concerne l'application. Dans cette thèse, nous nous sommes focalisés sur le contexte où l'on s'intéressait à la diffusion d'un seul fichier. Il pourrait être intéressant d'étendre notre étude à celui où il y aurait plusieurs fichiers à télécharger. On se retrouverait alors, comme nous l'avons signalé à plusieurs reprises dans ce manuscrit, dans une configuration proche du peer-to-peer. Les algorithmes utilisés dans un tel contexte pourraient alors être repris et étendus au travers des résultats que nous avons mis en évidence dans notre travail. Bien évidemment, nos travaux gardent tout leur sens si l'on change le paradigme de transmission des données et qu'au lieu de se fonder sur l'utilisateur, on se fonde sur l'information.

Le deuxième axe d'extension provient du contexte dans lequel nous avons mené nos expérimentations. Nous nous sommes focalisés sur des scénarios autoroutiers avec un cas de figure dans lequel tous les véhicules sont équipés en 802.11p. Des contextes urbains pourraient être envisagés même si l'on sait que la détermination de scénarios représentatifs est beaucoup plus complexe ainsi que l'analyse et l'interprétation des

7. CONCLUSIONS

résultats. Le fait que le taux de pénétration soit dans un premier temps relativement modeste n'a pas un impact fort sur les solutions **I2V** que nous avons proposées. En revanche, pour ce qui est du **V2V**, les travaux pourraient être repris et étendus.

Le troisième type d'extension porte sur les mécanismes que nous avons proposés. Parmi ceux-ci, on peut en particulier s'intéresser au codage de l'information. Nous avons mis en évidence tout l'intérêt que l'on avait à utiliser le codage réseau dans la partie **I2V**. Il s'est révélé très performant mais il est connu pour être coûteux en temps de calcul, on peut donc envisager d'autres techniques de codages tels que les Codes Fontaines pour le remplacer. Par exemple, nous pourrions déterminer la faisabilité de l'implantation du RaptorQ dans l'application proposée.

Notre analyse de l'état de l'art avait montré que de nombreuses propositions avaient été faites pour les communications **V2V** en particulier au niveau applicatif. Nous nous sommes donc concentrés simplement sur la complémentarité entre **V2V** et **I2V**. Nous avons analysé quelques stratégies simples sur un scénario où la zone de couverture des **RSU** est tout de même assez grande. Nous pourrions encore réduire le temps et les ressources nécessaires pour télécharger un fichier. Pour cela, on pourrait avoir recours à des mécanismes situés à plusieurs niveaux protocolaires en essayant de les combiner. Il s'agirait par exemple d'algorithmes épidémiques de dissémination[30] ou bien encore de routage opportuniste[69–74]. Des approches mêlant routage opportuniste et **NC** ont déjà été proposées, elles pourraient être testées dans notre contexte.

Dans notre travail, nous nous sommes finalement concentrés sur la technologie **WAVE**. Les travaux récents du 3GPP concernant l'« **evolved Multimedia Broadcast Multicast Service (eMBMS)** » pour le LTE refont gagner de l'intérêt pour la diffusion sur une infrastructure de réseaux télécoms. Dans le cas où **WAVE** serait saturé par les applications de sécurité routière, cette solution pourrait se révéler très utile. La complémentarité entre les approches que nous avons développées dans cette thèse et l'**eMBMS** est de notre point de vue particulièrement prometteuse. En termes de solutions télécoms à bas coût, les évolutions récentes du WiFi sont également à étudier de près. Les débits annoncés sont colossaux et l'on pourrait tout à fait envisager que la partie **I2V** soit prise en compte par ce biais. Nos travaux et nos idées s'appliqueraient alors pratiquement directement.

Références

- [1] ETSI, *GEO-Mobile Radio Interface Specifications (Release 3); Third Generation Satellite Packet Radio Service; Part 1 : General specifications; Sub-part 3 : General System Description; GMR-1 3G 41.202*, ETSI Std., Rév. ETSI TS 101 376-1-3, 2009.
- [2] —, *Digital Video Broadcasting (DVB); Framing Structure, channel coding and modulation for Satellite Services to Handheld devices (SH) below 3 GHz*, ETSI Std., Rév. ETSI EN 302 583, 2008.
- [3] V. Soares, F. Farahmand, et J. Rodrigues, “A layered architecture for Vehicular Delay-Tolerant Networks,” dans *IEEE Symposium on Computers and Communications, ISCC*, 2009, p. 122 –127.
- [4] IEEE, *IEEE Standard for Information Technology - Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN)–Specific requirements Part 22 : Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications : Policies and Procedures for Operation in the TV Bands*, IEEE Std., 2011.
- [5] —, *IEEE Standard for Air Interface for Broadband Wireless Access Systems*, IEEE Std., 2012.
- [6] —, *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., 2012.
- [7] —, *IEEE Standard for Information technology - Local and metropolitan area networks– Specific requirements– Part 11 : Wireless LAN Medium Access Control*

RÉFÉRENCES

- (MAC) and Physical Layer (PHY) Specifications Amendment 5 : Enhancements for Higher Throughput, IEEE Std., 2009.
- [8] —, *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 Amendment 6 : Wireless Access in Vehicular Environments*, IEEE Std., 2010.
- [9] ETSI, *Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*, ETSI Std., Rév. 1.1.0, 2010.
- [10] Z. D. Chen, H. Kung, et D. Vlah, “Ad hoc relay wireless networks over moving vehicles on highways,” dans *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01. ACM, 2001, p. 247–250.
- [11] J. Ott et D. Kutscher, “Drive-thru internet : IEEE 802.11b for "automobile" users,” dans *Proceedings of the 23th IEEE International Conference on Computer Communications, INFOCOM*, vol. 1, 2004, p. 362–373.
- [12] A. Chen, B. Khorashadi, D. Ghosal, et C.-N. Chuah, “Impact of transmission power on TCP performance in vehicular ad hoc networks,” dans *Fourth Annual Conference on Wireless on Demand Network Systems and Services, WONS '07*, 2007, p. 65–71.
- [13] K. Xu, S. Bae, S. Lee, et M. Gerla, “TCP behavior across multihop wireless networks and the wired internet,” dans *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia, WOWMOM '02*. ACM, 2002, p. 41–48.
- [14] A. Nandan, S. Das, G. Pau, M. Gerla, et M. Sanadidi, “Co-operative downloading in vehicular ad-hoc wireless networks,” dans *Second Annual Conference on Wireless On-demand Network Systems and Services, WONS*, 2005, p. 32 – 41.
- [15] J. Zhao et G. Cao, “VADD : Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks,” dans *Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM*, 2006, p. 1 –12.
- [16] B. B. Chen et M. C. Chan, “MobTorrent : A Framework for Mobile Internet Access from Vehicles,” dans *Proceedings of the 28th IEEE International Conference on Computer Communications, INFOCOM*, 2009, p. 1404 –1412.

- [17] F. Malandrino, C. Casetti, C.-F. Chiasserini, et M. Fiore, “Content downloading in vehicular networks : What really matters,” dans *Proceedings of the 30th IEEE International Conference on Computer Communications, INFOCOM*, 2011, p. 426–430.
- [18] —, “Optimal Content Downloading in Vehicular Networks,” *IEEE Transactions on Mobile Computing*, vol. 12, p. 1377–1391, 2013.
- [19] D. Zhang et C. Yeo, “Enabling Efficient WiFi-Based Vehicular Content Distribution,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, p. 479–492, 2013.
- [20] O. Trullols-Cruces, M. Fiore, et J. Barcelo-Ordinas, “Cooperative Download in Vehicular Environments,” *IEEE Transactions on Mobile Computing*, vol. 11, p. 663–678, 2012.
- [21] M. Sardari, F. Hendessi, et F. Fekri, “Infocast : A New Paradigm for Collaborative Content Distribution from Roadside Units to Vehicular Networks,” dans *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON '09.*, 2009, p. 1–9.
- [22] O. Trullols-Cruces, J. Morillo-Pozo, J. Barcelo, et J. Garcia-Vidal, “A Cooperative Vehicular Network Framework,” dans *IEEE International Conference on Communications, ICC '09*, 2009, p. 1–6.
- [23] W. Yuen, R. Yates, et S.-C. Mau, “Exploiting data diversity and multiuser diversity in noncooperative mobile infostation networks,” dans *Proceedings of the 22th IEEE International Conference on Computer Communications, INFOCOM*, vol. 3, 2003, p. 2218 – 2228 vol.3.
- [24] M. Fiore, C. Casetti, et C.-F. Chiasserini, “On-demand content delivery in vehicular wireless networks,” dans *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, MSWiM '05*. ACM, 2005, p. 87–94.
- [25] M. Fiore, C. Casetti, C.-F. Chiasserini, et M. Garetto, “Analysis and simulation of a content delivery application for vehicular wireless networks,” *Journal on Performance Evaluation*, vol. 64, p. 444 – 463, 2007.

RÉFÉRENCES

- [26] M. Fiore, F. Mininni, C. Casetti, et C.-F. Chiasserini, “To Cache or Not To Cache?” dans *Proceedings of the 28th IEEE International Conference on Computer Communications, INFOCOM*, 2009, p. 235–243.
- [27] H. Wu, R. Fujimoto, R. Guensler, et M. Hunter, “MDDV : a mobility-centric data dissemination algorithm for vehicular networks,” dans *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, VANET '04*. ACM, 2004, p. 47–56.
- [28] J. Zhao, Y. Zhang, et G. Cao, “Data Pouring and Buffering on the Road : A New Data Dissemination Paradigm for Vehicular Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 56, p. 3266–3277, 2007.
- [29] A. Skordylis et N. Trigoni, “Delay-bounded routing in vehicular ad-hoc networks,” dans *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '08*. ACM, 2008, p. 341–350.
- [30] M. Nekovee, “Epidemic algorithms for reliable and efficient information dissemination in vehicular,” *Intelligent Transport Systems, IET*, vol. 3, p. 104–110, 2009.
- [31] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, et S. Wicker, “An empirical study of epidemic algorithms in large scale multihop wireless networks,” Technical Report IRB-TR-02-003, Intel Research, Tech. Rep., 2002.
- [32] F. Ye, S. Roy, et H. Wang, “Efficient Inter-Vehicle Data Dissemination,” dans *IEEE Vehicular Technology Conference, VTC Fall*, 2011, p. 1–5.
- [33] —, “Efficient Data Dissemination in Vehicular Ad Hoc Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 30, p. 769–779, 2012.
- [34] J. Burgess, B. Gallagher, D. Jensen, et B. N. Levine, “MaxProp : Routing for Vehicle-Based Disruption-Tolerant Networks,” dans *Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM*, 2006, p. 1–11.
- [35] J. Isento, J. Dias, J. Neves, V. Soares, J. Rodrigues, A. Nogueira, et P. Salvador, “FTP@VDTN 2014 ; A file transfer application for Vehicular Delay-Tolerant Networks,” dans *IEEE International Conference on Computer as a Tool, EUROCON*, 2011, p. 1–4.

- [36] J. Dias, J. Isento, M. Pereira, B. Silva, V. Soares, J. Rodrigues, P. Pereira, A. Casaca, C. Cervello-Pastor, et J. Gallego, “WWW@VDTN - A Web browsing application for Vehicular Delay-Tolerant Networks,” dans *IEEE 16th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*, 2011, p. 11 –15.
- [37] S. Das, A. Nandan, et G. Pau, “SPAWN : a swarming protocol for vehicular ad-hoc wireless networks,” dans *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, VANET '04*. ACM, 2004, p. 93–94.
- [38] K. Lee, S.-H. Lee, R. Cheung, U. Lee, et M. Gerla, “First Experience with CarTorrent in a Real Vehicular Ad Hoc Network Testbed,” dans *Mobile Networking for Vehicular Environments*, 2007, p. 109 –114.
- [39] K. Lee et I. Yap, “CarTorrent : A Bit-Torrent System for Vehicular Ad-hoc Networks,” *Los Angeles*, 2007.
- [40] M. Abuelela et S. Olariu, “Zipper : a zero-infrastructure peer-to-peer system forvanet,” dans *Proceedings of the 3rd ACM workshop on Wireless multimedia networking and performance modeling, WMuNeP '07*. ACM, 2007, p. 2–8.
- [41] R. Ahlswede, N. Cai, S.-Y. Li, et R. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, p. 1204 –1216, 2000.
- [42] C. Gkantsidis, J. Miller, et P. Rodriguez, “Comprehensive view of a live network coding P2P system,” dans *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, IMC '06*. ACM, 2006, p. 177–188.
- [43] G. Ma, Y. Xu, K. Ou, et W. Luo, “How Can Network Coding Help P2P Content Distribution ?” dans *IEEE International Conference on Communications, ICC '09.*, 2009, p. 1 –5.
- [44] M. Li, Z. Yang, et W. Lou, “CodeOn : Cooperative Popular Content Distribution for Vehicular Networks using Symbol Level Network Coding,” *IEEE Journal on Selected Areas in Communications*, vol. 29, p. 223 –235, 2011.
- [45] Q. Yan, M. Li, Z. Yang, W. Lou, et H. Zhai, “Throughput Analysis of Cooperative Mobile Content Distribution in Vehicular Network using Symbol Level Network Coding,” *IEEE Journal on Selected Areas in Communications*, vol. 30, p. 484 –492, 2012.

RÉFÉRENCES

- [46] R. Koetter et M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Transactions on Networking*, vol. 11, p. 782 – 795, 2003.
- [47] S. Ahmed et S. S. Kanhere, “VANETCODE : network coding to enhance cooperative downloading in vehicular ad-hoc networks,” dans *Proceedings of the 2006 international conference on Wireless communications and mobile computing, IWCMC '06*. ACM, 2006, p. 527–532.
- [48] U. Lee, J.-S. Park, J. Yeh, G. Pau, et M. Gerla, “Code torrent : content distribution using network coding in VANET,” dans *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking, MobiShare '06*. ACM, 2006, p. 1–5.
- [49] M. Johnson, L. D. Nardis, et K. Ramch, “Collaborative Content Distribution for Vehicular Ad Hoc Networks,” *Allerton Conference on Communication, Control, and Computing*, p. 751–760, 2006.
- [50] A. Afanasyev, S. Lee, U. Lee, et M. Gerla, “Content Distribution in VANETs using Network Coding : Comparison of Generation Selection Mechanisms,” UCLA Computer Science Department, Tech. Rep., 2009, non publié. [Online]. Available : <http://lasr.cs.ucla.edu/afanasyev/data/files/Afanasyev/ContentDistributioninVANETsusingNetwork.pdf>
- [51] R. Jain, D.-M. Chiu, et W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation, 1984.
- [52] D. Stauffer et A. Aharony, *Introduction To Percolation Theory*. CRC Press, 1994.
- [53] Y. Chen, G. Paul, R. Cohen, S. Havlin, S. P. Borgatti, F. Liljeros, et H. E. Stanley, “Percolation theory and fragmentation measures in social networks,” *Physica A : Statistical Mechanics and its Applications*, vol. 378, p. 11 – 19, 2007.
- [54] A. Ksentini et M. Ibrahim, “Modeling and performance analysis of an improved DCF-based mechanism under noisy channel,” dans *Fourth International Conference on Broadband Communications, Networks and Systems, BROADNETS*, 2007, p. 874 –879.
- [55] X. Wang, “Performance modeling of IEEE 802.11 DCF using equilibrium point analysis,” dans *20th International Conference on Advanced Information Networking and Applications, AINA*, vol. 1, 2006, p. 6.

- [56] Y. Zheng, K. Lu, D. Wu, et Y. Fang, “Performance Analysis of IEEE 802.11 DCF in Imperfect Channels,” *IEEE Transactions on Vehicular Technology*, vol. 55, p. 1648 –1656, 2006.
- [57] Y. Lee, M. Y. Chung, et T.-J. Lee, “Performance Analysis of IEEE 802.11 DCF under Nonsaturation Condition,” *Mathematical Problems in Engineering*, 2008.
- [58] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE Journal on Selected Areas in Communications*, vol. 18, p. 535 – 547, 2000.
- [59] IEEE, *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., 2007.
- [60] J.-Y. Le Boudec et M. Vojnovic, “Perfect simulation and stationarity of a class of mobility models,” dans *Proceedings of the 24th IEEE International Conference on Computer and Communications Societies, INFOCOM*, vol. 4, 2005, p. 2743–2754 vol. 4.
- [61] ASTM, *Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems — 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ASTM Std.
- [62] H. Arbabi et M. Weigle, “Highway mobility and vehicular ad-hoc networks in ns-3,” dans *Proceedings of the 2010 Winter Simulation Conference, WSC*, 2010, p. 2991 –3003.
- [63] M. Treiber, “Dynamic traffic simulation,” <http://traffic-simulation.de/MOBIL.html>, 2010.
- [64] G. Pei et T. R. Henderson, “Validation of OFDM error rate model in ns-3,” Boeing Research & Technology, Tech. Rep., 2010.
- [65] P. A. Chou, Y. Wu, et K. Jain, “Practical network coding,” dans *Proceedings of the annual Allerton conference on communication control and computing*, vol. 41. The University ; 1998, 2003, p. 40–49.

RÉFÉRENCES

- [66] C.-C. Chao, C.-C. Chou, et H.-Y. Wei, “Pseudo Random Network Coding Design for IEEE 802.16m Enhanced Multicast and Broadcast Service,” dans *IEEE 71st Vehicular Technology Conference, VTC Spring*, 2010, p. 1–5.
- [67] S.-Y. Li, R. W. Yeung, et N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, p. 371–381, 2003.
- [68] T. Ho, R. Koetter, M. Medard, D. Karger, et M. Effros, “The benefits of coding over routing in a randomized setting,” dans *Proceedings of the IEEE International Symposium on Information Theory*, 2003, p. 442.
- [69] S. Biswas et R. Morris, “ExOR : opportunistic multi-hop routing for wireless networks,” dans *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05. ACM, 2005, p. 133–144.
- [70] H. Dubois-Ferriere, M. Grossglauser, et M. Vetterli, “Least-cost opportunistic routing,” dans *Proceedings of 2007 Allerton Conference on Communication, Control, and Computing*, vol. 11, 2007, p. 13–39.
- [71] R. Shah, S. Wietholter, A. Wolisz, et J. Rabaey, “When does opportunistic routing make sense ?” dans *Third IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom 2005 Workshops.*, 2005, p. 350–356.
- [72] Y. Lin, B. Li, et B. Liang, “CodeOR : Opportunistic routing in wireless mesh networks with segmented network coding,” dans *IEEE International Conference on Network Protocols, ICNP*, 2008, p. 13–22.
- [73] Y. Lin, B. Liang, et B. Li, “SlideOR : Online Opportunistic Network Coding in Wireless Mesh Networks,” dans *Proceedings of the 29th IEEE International Conference on Computer Communications, INFOCOM*, 2010, p. 1–5.
- [74] D. Koutsonikolas, C.-C. Wang, et Y. Hu, “CCACK : Efficient Network Coding Based Opportunistic Routing Through Cumulative Coded Acknowledgments,” dans *Proceedings of the 29th IEEE International Conference on Computer Communications, INFOCOM*, 2010, p. 1–9.

Liste des publications

Conférences Internationales

- [75] Darwin Astudillo, Emmanuel Chaput, et Andre-Luc Beylot, “Improving performance of Map Updates through Satellite Communications in Vehicular Networks,” dans *28th AIAA International Communications Satellite Systems Conference, ICSSC*. American Institute of Aeronautics and Astronautics, Août 2010.
- [76] —, “Map update application : Performance measurements on a highway scenario,” dans *International Conference on Wireless Communications in Unusual and Confined Areas, ICWCUCA*, Août 2012.
- [77] —, “Bulk data transfer using VANET infrastructure,” dans *IEEE 77th Vehicular Technology Conference, VTC Spring*, Juin 2013.

Ateliers Nationaux

- [78] Darwin Astudillo, Emmanuel Chaput, et Andre-Luc Beylot, “Map Updates in Vehicular Networks with different technologies,” *Pôle ResCom Summer School, Giens*, Juin 2010.