
Expérimentations des communications inter-véhicules

Sofiane Khalfallah^{*} — Moez Jerbi^{} — Mohamed Oussama Cherif^{**} — Sidi-Mohammed Senouci^{**} et Bertrand Ducourthial^{*}**

Lab. Heudiasyc UMR CNRS 6599 ^{*}
Université de Technologie de Compiègne
Compiègne, France
^{*} @utc.fr

Orange Labs ^{**}
2 Avenue Pierre Marzin
22307 Lannion Cedex, France
^{**} @orange-ftgroup.com

RÉSUMÉ. Actuellement, les systèmes de transport intelligent (ITS) sont de plus en plus étudiés aussi bien dans la communauté de recherche que dans les industries (automobiles, télécommunications, etc.). En effet, la mise en place d'applications ITS peut à court terme améliorer la sécurité routière, augmenter efficacement l'utilisation des routes, réduire les congestions et les embouteillages, limiter l'impact des véhicules sur l'environnement, etc. Cependant, plusieurs problématiques doivent être résolues avant de pouvoir concevoir de telles applications, parmi lesquelles les communications inter-véhicules ont une place privilégiée. Dans cet article, nous nous intéressons aux performances que peuvent offrir les réseaux de véhicules multisauts, et ce dans différents scénarios, et à différents niveaux. Dans un premier temps, nous évaluons les performances du protocole 802.11 utilisé avec un routage IP statique. Dans un second temps, nous évaluons les performances d'un protocole de routage dédié aux VANETs et développé en mode utilisateur dans une architecture adaptée aux réseaux dynamiques. Notre démarche est expérimentale et deux plates-formes de tests ont été développées afin d'expérimenter sur route jusqu'à six véhicules. Ces premiers résultats donnent une idée de ce que les futures applications pour réseaux de véhicules disposeront en terme de délai ou de débit.

MOTS-CLÉS : réseaux de véhicules, VANET, réseaux ad hoc dynamiques, ITS, IEEE 802.11, plates-formes expérimentales, évaluation de performances.

1. Introduction

Les communications inter-véhicules constituent un domaine de recherche récent, qui intéresse de plus en plus la communauté scientifique, les constructeurs automobiles et les opérateurs des télécommunications. En effet, les systèmes de communication inter-véhicules (IVC) peuvent être utilisés pour mettre en place plusieurs types d'applications. Ces applications sont connues sous le nom de systèmes de transport intelligents (ITS). Ces systèmes permettent d'améliorer les moyens de transport que se soit en terme de sécurité routière, de mobilité, d'impact sur l'environnement, de productivité, etc. Les applications ITS peuvent être à la fois mises en place dans l'infrastructure des routes et dans les véhicules eux-mêmes. Ce type de fonctionnement confère aux applications ITS trois types de déploiements : (i) véhicule-vers-véhicule (V2V), (ii) infrastructure-vers-véhicule (I2V) et véhicule-vers-infrastructure (V2I).

Dans [DUC 07a], les applications ITS, sont classées en quatre catégories : (i) applications orientées infrastructures afin d'optimiser leur gestion (gestion des autoroutes, transport de marchandises, organisation des urgences etc.), (ii) applications orientées véhicules afin d'améliorer la sécurité routière (alertes accidents, prévention des carambolages, conduite assistée etc.), (iii) applications orientées conducteurs afin de faciliter l'usage de la route (signalisation dynamique pour les embouteillages ou les travaux, estimation temps réelle du meilleur trajet, etc.), (iv) applications orientées

passagers pour offrir de nouveaux services à bord de la voiture (accès internet, jeux distribués, messagerie instantanée, informations touristiques *etc.*).

Plusieurs problématiques doivent être résolues avant de pouvoir concevoir de telles applications. En effet, les applications ITS exigent dans la majorité des cas des communications robustes, une certaine qualité de service ou des communications temps réelles. Or, ceci contraste avec les réseaux de véhicules qui présentent des inconvénients majeurs dus à leur forte dynamique (perte fréquente de connectivité, communication non fiable, durées de communications non bornée *etc.*). Différents travaux de recherche ont été proposés récemment pour faire face à certaines de ces problématiques [ABD 07, DUC 07a]. De nombreux protocoles de routage, de localisation ou d'adressage ont été étudiés [KHA 07b]. Des projets ont défini des architectures logicielles pour réaliser les communications véhicules-véhicules ou véhicules-infrastructures. Dans la plupart des cas, les protocoles ont été évalués par le biais de simulations [KHA 05]. Cependant les simulateurs actuels peinent à modéliser les conditions de tests sur route.

Dans cet article, nous relatons les expériences sur routes que nous avons menées afin d'évaluer les performances qu'il est possible d'attendre des communications inter-véhicules. Pour cela, nous avons mis en œuvre deux plates-formes de tests complémentaires, appelées Carman et Caremba. La première plate-forme est dédiée à la mesure de performances des couches réseaux dans les réseaux de véhicules. La deuxième plate-forme mesure les performances applicatives obtenues avec un protocole de routage dédié aux réseaux de véhicules [KHA 07b] et développé en espace utilisateur sur une architecture adaptée aux réseaux dynamiques [DUC 07a]. L'analyse des premiers résultats donne une idée de ce que les futures applications ITS pourront disposer en terme de débits ou de délais notamment.

La suite de l'article est structurée de la façon suivante. Dans un premier temps, nous résumons les travaux concernant les expériences sur route ainsi que les architectures embarquées sous-jacentes aux applications ITS. Ensuite, nous détaillons l'architecture logicielle et matérielle de nos deux plates-formes expérimentales ainsi que les scénarios de tests avant d'analyser les premiers résultats obtenus. Nous terminons par quelques perspectives.

2. État de l'art sur les expérimentations

Différentes expérimentations ont concerné les VANETs. Elles utilisent principalement la norme IEEE 802.11. Dans [ABD 07], les auteurs démontrent en effet la faisabilité d'utiliser directement la norme IEEE 802.11b d'une voiture en mouvement connecté à un point d'accès dans le désert de Californie, où il n'y a aucun obstacle et aucune interférence radio et aucun autre véhicule. Dans [OTT 04], les auteurs ont étudié le comportement des connexions réseaux (TCP et UDP) pour une voiture en mouvement, connectée à un point d'accès. Le but était de comprendre l'impact de la vitesse de la voiture, du taux de transmission et de la taille des paquets. Dans ces expérimentations, des cartes IEEE 802.11b ont été utilisées et peu de précisions ont été révélées sur l'architecture embarquée utilisée.

Dans [BUC 05], les auteurs présentent quelques résultats expérimentaux utilisant une application multimédia dans un réseau ad hoc inter-véhicules. Ils ont utilisé deux véhicules équipés de la norme IEEE 802.11b et ont fait un ensemble de tests dans deux environnements différents (urbain et autoroute). Les auteurs arrivent aux conclusions suivantes : (i) le rapport Signal/Bruit (SNR pour *Signal to Noise Ratio*) est plus important sur autoroute que dans une zone urbaine, (ii) la liaison est plus fiable sur autoroute que dans une ville et (iii) il vaut mieux utiliser de grands paquets de données sur autoroute et de petits paquets (grande fragmentation) en zone urbaine. L'architecture utilisée repose sur Linux, des cartes PCMCIA, des GPS et le protocole UDP. Des résultats similaires sont obtenus.

nus dans [HUI 01] quant au SNR et au bruit. De plus, les auteurs analysent d'autres paramètres de performances tels que le RTT, le débit TCP et UDP, *etc.* Ils ont utilisé trois véhicules et un routage statique. Selon les auteurs, le déploiement d'applications multimédia est difficile dans un réseau de véhicules multi-saut. Leur architecture embarquée repose sur Windows XP, les protocoles UDP et TCP, iperf [IPE] pour la génération du trafic, un GPS, Netstumbler pour la mesure des signaux et un routage statique sur trois nœuds. Dans [SIN 02], les auteurs ont mesuré la qualité du lien sur autoroute dans des environnements urbains et semi-urbains. Les résultats de l'étude ont montré que l'environnement semi-urbain est le plus favorable pour la communication inter-véhicules. L'architecture utilisée par ces auteurs utilise Linux, un GPS, le protocole UDP et l'outil d'évaluation de performances réseau Netperf.

Le but des expériences faites par [GAS 06, OTT 04] est de comprendre les performances en termes de temps de connexion et de taux de perte quand une voiture mobile se connecte à des points d'accès. L'architecture embarquée utilisée dans [GAS 06] utilise Linux, les protocoles TCP, UDP et HTTP, iperf pour la génération du trafic TCP, wget et apache pour la génération de trafic HTTP et enfin tcpdump (couche réseau) et kismet (couche liaison) pour l'acquisition des données.

Ces divers expériences ne considèrent pas des scénarios de tests proches de la réalité (*i.e.*, croisement, dépassement ou multi-sauts entre les véhicules dans des scénarios hybrides tel que V2V et I2V). Un autre point important qui n'a pas été suffisamment traité est celui de l'influence de certains facteurs tel que la vitesse, l'accélération ou la distance sur les métriques de performance. Enfin, les expériences présentées dans cet article utilisent des routages statiques. Elles ne prennent pas en compte des protocoles de routage spécifiques aux réseaux de véhicules, ni des architectures embarquées adaptées aux applications devant évoluer en environnement dynamique. Or, les applications ITS reposeront très certainement sur des architectures logicielles spécifiques, intégrant des piles protocolaires adaptées. Divers projets de recherche portent sur les applications ITS aux USA (VII, CICAS, IVBSS...), en Europe (CVIS, SAFESPOT, COOPERS, PReVENT, GST, HIGHWAY, FLEETNET...), au Japon (SmartWay, VICS), en Inde (ITSIndia), en Allemagne (NOW), en France (PREDIT) *etc.* Actuellement, peu de projets ont spécifié leurs architectures embarquées ou publié des documents publics les concernant. L'architecture proposée par le projet GST (qui concerne les communications véhicules-infrastructure) se base sur Linux, OSGi [OSG], IPv6, HTTP, SOAP et le protocole OMA DM [GST]. L'architecture du projet CVIS [CVI] est basée sur Linux, OSGi, IPv6 et CALM. Plusieurs protocoles de routage spécifiques aux réseaux de véhicules sont à l'étude dans ces projets. Notons que les architectures devront rester ouvertes à de nouvelles solutions car le domaine est jeune, et la dynamique des réseaux de véhicules remet en cause de nombreuses solutions existantes, depuis les couches basses jusqu'aux applications en passant par la sécurité.

Cet article se propose de mesurer les performances des réseaux de véhicules dans divers scénarios réels sur route, avec et sans architecture et protocole spécifiques aux réseaux de véhicules.

3. Expérimentations avec la norme IEEE 802.11 sur route

Afin d'étudier les performances de la norme IEEE 802.11 et de mieux appréhender les différents problèmes qui surviennent lors de la communication inter véhicules, nous avons procédé à de nombreux tests en condition réelle. Pour ce faire, nous avons mis en place notre propre plate-forme de test Caremba. Cette plate-forme est assez générique et permet de réaliser des tests sur route sur des réseaux de deux à six voitures, et de procurer les métriques permettant de valider ou non les scénarios de tests réalisés (*e.g.*, qualité du signal pour chaque trame émise sur chaque nœud, mesure du bruit, distance entre véhicules, vitesse et direction des véhicules, taux de perte, débit mesuré, quantité des données reçues, durée de connexion, gigue, temps d'aller retour).

3.1. Description de la plate-forme Carman

Lors des expérimentations, nous avons utilisé des voitures équipées du matériel présenté dans la figure 1. La plate-forme Carman (CAR-based Mobile Ad hoc Network) fonctionne sous Linux et a



Figure 1. Matériel utilisé.

été testée sur plusieurs distributions. En effet, suivant la distribution utilisée, certains outils réseaux sont nécessaires (*e.g.*, *ifconfig*, *ping*, les extensions et outils de Jean Tourrilhes [TOU], le package *Bluez* pour la connexion du GPS, *etc.*). Il faut noter également que la version du pilote *madwifi* [MAD] a été modifiée afin de permettre la capture d'informations avancées telles que la qualité du signal et le niveau de bruit. Plusieurs logiciels sont indispensables au fonctionnement de la plate-forme tels que le logiciel *Iperf* pour la génération de trafic réseau TCP ou UDP, VideoLan pour la visualisation et la diffusion des vidéos, GNU Plot [PLO] pour la création de courbes de résultats, *etc.* La figure 2 permet de visualiser l'architecture de Carman. Cette architecture met en exergue les

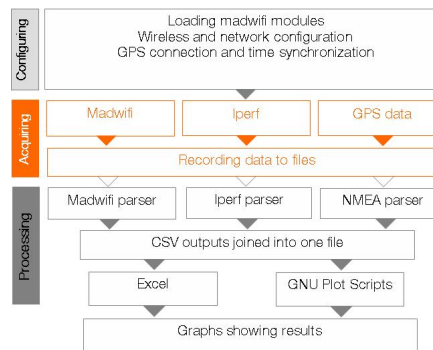


Figure 2. Architecture de Carman.

trois modules de cette plate-forme à savoir (i) un module pour la configuration de la plate-forme qui permet de charger les modules *madwifi*, de fixer les paramètres du réseau et de synchroniser les temps GPS/PC, (ii) un module pour l'acquisition des données qui est assez simple d'utilisation car le conducteur du véhicule peut aussi l'utiliser et enfin (iii) un module qui permet d'analyser les données capturées.

Au lancement du module d'acquisition des données, les différentes informations sont enregistrées dans des fichiers de logs. Au cours des expériences, les informations concernant les différentes métriques telles que le taux de perte et la qualité de signal sont affichées comme le montre la figure 3.

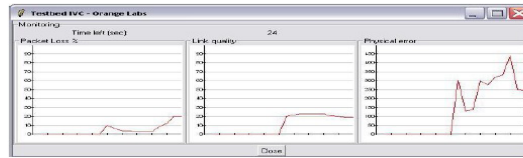


Figure 3. *Interface de moniteur.*

Dans le module d'analyse des données capturées, nous avons développé quatre analyseurs de trace (pour GPS, madwifi, Iperf et ping). Un script global a été créé pour appeler l'ensemble des analyseurs et fusionner les résultats. Ensuite le tracé des courbes est effectué grâce à GNU Plot.

3.2. Scénarios des tests

Les scénarios de tests envisagés ont pour but de couvrir le maximum des situations réelles. Nous allons évoquer les plus pertinents d'entre eux.

3.2.1. Scénarios V2V

Le premier scénario est le croisement de deux véhicules mobiles comme le montre la figure 4 (a) nous avons fait le test à faible vitesse 30 km/h et à vitesse modérée 50 km/h. Le scénario de

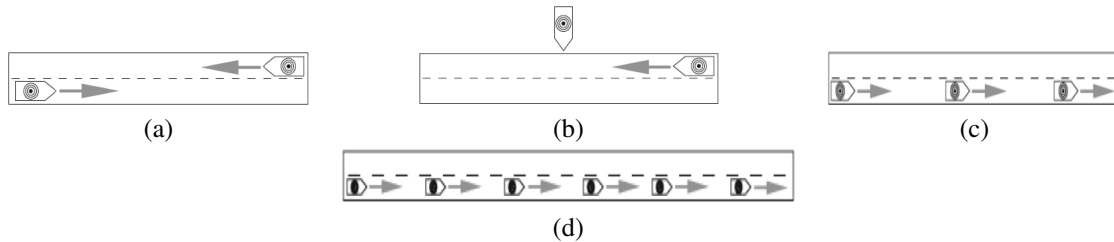


Figure 4. (a) *Croisement de deux véhicules*, (b) *Scénario de dépassement* (b), (c) *Scénario à trois véhicules*, (d) *Scénario à 6 véhicules*.

dépassement consiste au croisement de deux véhicules dont un est immobile (cf. figure 4 (b)). C'est une situation assez courante notamment dans les carrefours.

Le scénario d'accélération consiste à côtoyer deux véhicules puis faire accélérer l'un d'entre eux. Ceci permet de voir l'influence de l'accélération sur la communication.

Le scénario du simple saut est un scénario à trois véhicules qui permet de voir l'influence du passage par un nœud intermédiaire (cf. figure 4 (c)). Cette expérience a été faite avec des véhicules immobiles puis en mouvement avec deux vitesses prédéterminées (50 Km/h et 90 Km/h).

Le dernier scénario V2V testé est un scénario à six véhicules (cf. figure 4 (d)). Ce scénario permet d'étudier le comportement des communications utilisant quatre relais. Le choix des relais est préfixé grâce à l'affectation de routes statiques.

3.2.2. Scénarios I2V

Ces scénarios sont très importants et concernent le cas d'une infrastructure fixe proposant des services à automobilistes/passagers dans une ville ou sur une autoroute. Le scénario testé est basé sur une borne statique située sur un pont alors qu'un véhicule circule sur une voie express passant en dessous. La borne émet des données de façon continue avec un débit maximal. Le véhicule, ne

connaissant pas le canal d'émission de la borne, est obligé de le découvrir via une recherche. Le but est de s'approcher au maximum des conditions réelles.

3.2.3. Scénario hybride

Ce scénario a pour objectif d'étendre la couverture d'une borne et représente un scénario hybride des deux précédents. Pour mettre en place ce scénario, la borne émettrice est située sur un pont alors que deux véhicules suffisamment proches pour pouvoir communiquer empruntent la voie express passant en dessous (cf. figure 5). Au début, la borne se connecte directement au premier véhicule et lui envoie les données. Lorsque celui-ci devient hors de portée, la borne tente de faire relayer les données par le second véhicule.

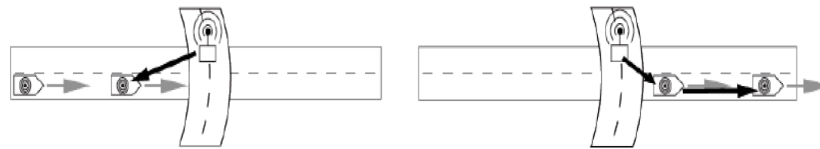


Figure 5. Scénario hybride à deux véhicules.

3.3. Résultats obtenus et analyses

3.3.1. Expériences V2V

Les résultats des scénarios V2V avec deux voitures sont illustrés par la figure 6 (dépassement et croisement). Grâce à ces scénarios, nous avons pu voir que la distance entre l'émetteur et le récepteur influe sur la qualité du signal et le taux de perte. Nous avons également pu constater que la vitesse affecte peu les performances des communications entre deux véhicules. La figure 7 montre un

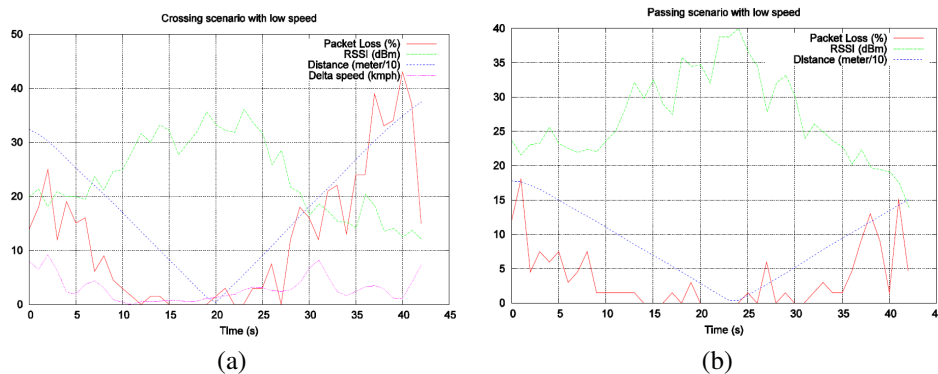


Figure 6. Scénario de croisement (a) et de dépassement (b) à petite vitesse

résumé des statistiques des premières expériences multi-sauts avec 3 voitures (statique et en suivi). En comparant les pertes moyennes obtenues et le délai d'aller-retour en petite ou grande vitesse, nous pouvons conclure que la vitesse affecte peu les performances des communications multi-sauts. On constate que les pertes dans le scénario de suivi sont inférieures au scénario statique malgré la mobilité de nœuds. Cette différence provient de l'environnement car sur autoroute, il n'y a ni obstacle ni interférence avec d'autres systèmes radios. La figure 8 résume les statistiques obtenues lors de la chaîne de communication de 6 véhicules, c'est-à-dire en utilisant 4 relais. Le taux de perte est de 15,69% de bout en bout soit environ 3,5% de perte par émission. En effet, les pertes

	Pertes moy (%)	RTT moy (ms)	Distance inter-vehicule (m)
Scénario statique	4,75	4,41	87,17
Scénario de suivi (PV)	1,99	1,97	91,39
Scénario de suivi (GV)	3,48	2,04	142,05

Figure 7. Résumé des statistiques des scénarios V2V

augmentent avec le nombre croissant de sauts radio. Ce phénomène s'explique par l'utilisation du mécanisme CSMA (plus le média est surchargé, plus la probabilité est grande que deux voitures émettent en même temps après avoir détecté un support libre augmentant ainsi le taux de pertes) et par le problème de station cachée (mécanisme RTS/CTS désactivé).

Distance inter véhicule m	71,93
Distance Total m	359,64
Vitesse km/h	78,69
Débit effectif kbps	252,67
Débit émetteur kbps	300
Taux de perte %	15,69
RTT ms	7,99
RSSI	42,71
Bruit dB	-98,79

Figure 8. Moyenne des métriques de la chaîne de 6 véhicules

3.3.2. Expériences V2I

Pour ces expérimentations, nous nous sommes focalisés sur les métriques suivantes : la puissance du signal, le taux de perte, la durée de connexion au réseau et le total de données reçues. Nous avons exécuté plusieurs fois les essais menant à des résultats semblables. La figure 9 représente la perte de paquet et le RSSI (mesure de la puissance reçue d'un signal radio) pour le scénario simple avec un véhicule et un point d'accès (a) et pour le scénario hybride avec deux véhicules et un point d'accès (b). L'impact des ponts est clairement visible. La figure 10 montre le temps de connexion et

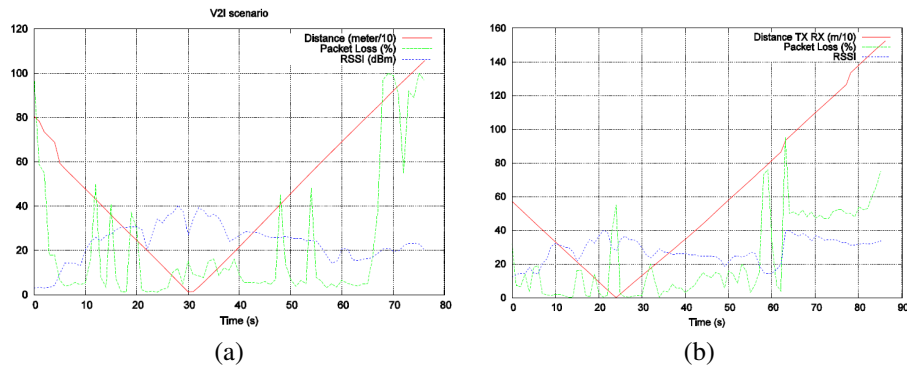


Figure 9. Taux de perte et RSSI pour le scénario simple (a) et hybride (b) (grande vitesse).

la quantité de données transférées pour les deux derniers scénarios simple et hybride. Les résultats expérimentaux ont révélé la faisabilité du réseau ad hoc pour étendre la zone de couverture des points

	Temps de connexion		Quantité de données	
	Grande vitesse	Petite vitesse	Grande vitesse	Petite vitesse
Total	81	140	59.195	98.86

(a)

	Temps de connexion		Quantité de données	
	Grande vitesse	Petite vitesse	Grande vitesse	Petite vitesse
Direct	66	143	54.247	106.452
Multihop	25	27	9.813	9.519
Total	91	170	64	116

(b)

Figure 10. *Durée de connexion (en secondes) et quantité de données (en méga-octets) transférées dans le scénario simple (a) et hybride (b)*

d'accès. Le réseau ad hoc accroît ainsi la durée de connexion pour des voitures en mouvement de plus de 16%. Il augmente également la quantité de données que la voiture peut recevoir de plus de 10%.

4. Expérimentations avec un protocole de routage et une architecture dédiée aux VANET

Dans cette section, nous complétons les expérimentations précédentes en présentant des tests sur route réalisés avec une architecture logicielle embarquée et un protocole de routage dédié aux réseaux de véhicules. Les mesures sont réalisées au niveau applicatif, grâce à une application qui génère du trafic et qui mesure précisément les délais de transmission.

4.1. Description de la plate-forme Caremba

La plate-forme Caremba est constituée d'un certain nombre de kits composés de PC (PC *shoebox* industriels 12V ou portables), d'un récepteur GPS, d'une carte IEEE 802.11 et d'une antenne WiFi externe pour les communications (*cf.* figure 11). Notons que les antennes utilisées pour les tests ne sont pas toutes identiques (*i.e.*, puissance d'émission différentes).



Figure 11. *Plate-forme de test : Caremba.*

Les PC utilisent Linux et l'architecture logicielle Airplug. La suite logicielle Airplug [DUC 07a] a été développée de manière à permettre les interactions intra ou inter véhicules entre applications embarquées tout en garantissant une grande robustesse face aux défaillances. Elle est constituée d'un programme cœur (*airplug*) qui gère les communications et sur lequel sont connectés divers processus (*cf.* figure12-a). Les communications locales (communications entre différents processus sur la même machine) sont réalisées par des tubes. Les communications distantes font appel à des sockets utilisant divers protocoles de communication (des couches 2, 3 et 4, ou des couches 2 uniquement). De cette manière, chaque processus possède son propre espace mémoire, et seul *airplug* accède au réseau, ce qui permet diverses optimisations (*pyggybacking*, ordonnancement...) et renforce la robustesse de part l'indépendance des processus développés en mode utilisateur. En outre, cette architecture n'impose aucun paradigme ni langage de programmation puisque les applications doivent

simplement savoir lire leur entrée standard et écrire sur leur sortie standard. Airplug propose un routage inter-applications adapté aux réseaux dynamiques. Les applications peuvent s'abonner aux flux d'information produites par leur consœurs, ou bien envoyer une requête et attendre une réponse. Les messages peuvent être envoyés localement et/ou dans le voisinage, à une application particulière ou à toutes les applications abonnées.

Par ailleurs, cette architecture permet le développement de protocoles en mode utilisateur, ainsi que la cohabitation de plusieurs protocoles (*e.g.*, routage géographique pour les alertes, routage dirigé par la source pour la connexion à un point d'accès infrastructure). Les expériences reportées dans cet article ont utilisé les *transmissions conditionnelles* [KHA 07b], implémentées sous la forme d'une application compatible Airplug appelée HOP (*cf.* figure 12-b). Pour ces expériences, les communications inter-véhicules ont utilisé des socket UDP. Les transmissions conditionnelles remplacent l'adresse de destination par des conditions telles que la distance à l'émetteur, la trajectoire, la zone géographique *etc.* Grâce à l'évaluation dynamique de ces conditions, les performances obtenues sont très bonnes comparativement aux adresses réseaux ou géographiques, dont la pérennité est faible dans un réseau dynamique.

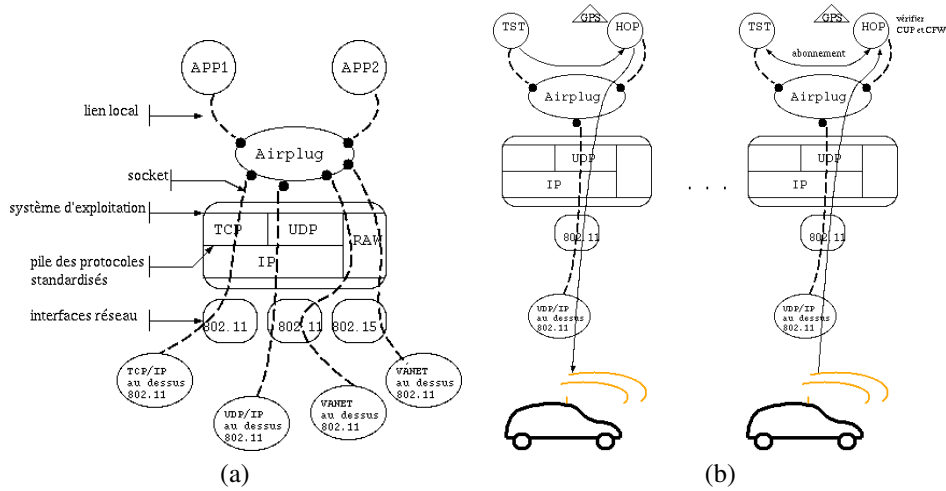


Figure 12. (a) Architecture embarquée Airplug, (b) Utilisation de HOP pour les tests.

4.2. Fonctionnement et scénarios

Pour le déroulement de ces expériences, une seule application TST (embarquée dans la voiture de tête) envoie des données (*cf.* figure 12 (b)). Cette application envoie localement à l'attention de HOP (via airplug) le message à transmettre et les conditions de transmission CUP et CFW. Les champs de contrôle CUP et CFW correspondent respectivement à la condition de réception du message, et à sa retransmission par l'application réceptrice. Une fois que l'application HOP reçoit ce message (via son entrée standard), elle envoie à airplug le message à diffuser. Le programme airplug diffuse alors ce message vers les autres applications HOP embarquées dans les autres voitures. Le message transmis contient les différents champs de contrôle dont les conditions CUP et CFW.

Dans les autres voitures, chaque application TST s'abonne à l'application HOP locale en lui envoyant (via airplug) un message de contrôle spécifique. Dès que le programme airplug reçoit un message, il vérifie l'application destinataire (ici HOP) et l'envoie en local. Une fois que l'application HOP récupère le message qui lui a été envoyé sur le réseau, elle vérifie dynamiquement les conditions

de transmission. Selon le résultat de ces évaluations, il transmet localement le message reçu aux applications abonnées à ses messages (ici TST) et/ou retransmet à nouveau ce message au réseau.

Dans ces expériences, les cinq véhicules sont en convoi (figure 13). Le programme TST de la voiture leader envoie à intervalle régulier des paquets de taille fixe. Il combine les datations GPS (reçues à la fréquence d'un hertz) avec les horloges locales des ordinateurs (non synchronisées) de manière à permettre une mesure fine des délais de transmission. Nous avons choisi des conditions pour CUP et CFW de telle sorte qu'on fixe une seule direction de transmission des messages (*i.e.*, voiture de tête vers voiture de queue). Si la voiture réceptrice est derrière l'émettrice alors HOP (*i*) transmet localement le message reçu à l'application TST abonnée à ses messages et (*ii*) retransmet à nouveau ce message dans le voisinage. Chaque applications HOP reçoit la position GPS *via* une application locale qui lit les trames NMEA envoyées par un récepteur GPS.

Les conditions CUP et CFW permettent de fixer une direction de transmission mais nous avons constaté que les messages ne faisaient pas tous autant de rebonds à cause des antennes hétérogènes et de l'environnement changeant. Par exemple, la troisième voiture recevait souvent des messages directement de la voiture leader (figure 13 haut). Dans une seconde expérience, nous avons contraint les messages à passer par toutes les voitures successives grâce aux conditions, même si la portée aurait permis d'éviter un saut (figure 13 bas).

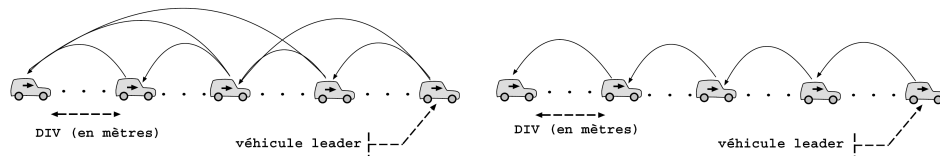


Figure 13. Scénarios.

4.3. Résultats et analyses

Dans le premier test, les résultats obtenus sont en terme de distance inter-véhicules (DIV). L'hétérogénéité des antennes explique en partie les résultats obtenus dans le tableau de la figure 14. En effet, la voiture 3 reçoit directement les paquets de la voiture 1 d'où un délai et un taux de pertes équivalents à la voiture 2. De plus, vu que la voiture 4 a une antenne de portée de communication inférieure à celle de la voiture 5, elle reçoit les paquets après la voiture 5. En effet, la voiture 5 reçoit les paquets aussi bien de la voiture 4 que de la 3 ou de la 2 et elle peut être considérée à 2 ou 3 sauts de la voiture 1 (d'où des délais largement inférieurs à ceux enregistrés dans la voiture 4). Notons aussi que la portée des antennes a été très variable en fonction du trafic et de l'environnement. En conclusion, il a été difficile d'obtenir des résultats en fonction de la distance.

Dans la deuxième expérience, en forçant les rebonds, nous avons obtenu des résultats en fonction du nombre de sauts. Le nombre de sauts impacte directement les performances de la réception (figure 15). Par exemple, le délai moyen de réception augmente linéairement (coefficient directeur de +246 ms).

5. Conclusion et Perspective

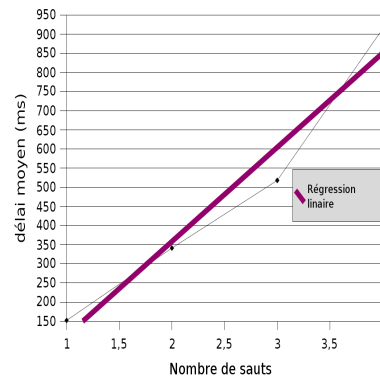
Dans cet article, nous nous sommes intéressés aux performances réelles des réseaux de véhicules. En utilisant deux plates-formes, nous avons obtenu des performances brutes avec le WiFi et un routage statique, et des performances applicatives avec un routage VANET.

PARAMETRES				
taille paquets ko	500			
total données émises ko	900.1			
voitures	2	3	4	5
DIV moyen m	57	103	182	268
pertes %	13	15	50	12
intervalles de pertes	279	311	667	274
délai 1 ^{er} paquet	971	979	1814	665
délai moyen ms	88	68	2722	884
total données reçues ko	778	767	453	788

Figure 14. Résultats obtenus selon la distance inter-véhicules DIV.

PARAMETRES				
taille paquets ko	500			
total données émises ko	1000			
nombre de sauts	1	2	3	4
pertes %	9	32	41	58
intervalles de pertes	97	207	184	364
délai 1 ^{er} paquet	95	132	260	386
délai moyen ms	154	341	518	914
total données reçues ko	694	636	652	578

(a)



(b)

Figure 15. (a) Résultats obtenus selon le nombre de sauts, (b) Régression linéaire (délai moyen)

La première plate-forme Carman nous a permis d'expérimenter au niveau des couches réseaux en utilisant un routage statique. Cette plate-forme reste assez facile à utiliser et permet de procurer plusieurs métriques nécessaires pour une caractérisation fine des communications dans les réseaux véhiculaires (*i.e.*, qualité du signal, mesure du bruit, taux de perte, débit, durée de connexion, gigue, temps d'aller retour *etc.*). La deuxième plate-forme nous a permis d'expérimenter un protocole de routage dédié aux VANET (les transmissions conditionnelles) dans un environnement applicatif adapté aux réseaux dynamiques (Airlug).

Ces deux plates-formes nous ont permis d'effectuer des tests mettant en place des réseaux allant de deux à six véhicules. Les premiers résultats obtenus lors de ces tests sont encourageants. En particulier, ils montrent la bonne performance de la norme 802.11 avec des pertes ou des délais relativement faible. Nous avons également constaté que la distance est un facteur de perte alors que la vitesse et l'accélération ne sont que peu influentes sur la communication. Les résultats expérimentaux ont révélé aussi la faisabilité du réseau ad hoc pour étendre la zone de couverture des points d'accès. Les premiers résultats obtenus au niveau applicatif sont encourageants. Le délai moyen après 5 sauts reste inférieur à la seconde. Par contre, le délai inter-paquets devra être ajusté afin de limiter les collisions dans le convoi. Ces travaux sont en cours.

Par la suite, nous prévoyons de compléter ces premiers résultats avec d'autres mesures et d'autres scénarios (communication de véhicules avec plusieurs points d'accès, évaluation des performances du handover *etc.*), d'évaluer les performances d'autre protocoles de routage et d'auto-organisation conçus pour les réseaux de véhicules (*e.g.*, GyTAR [JER 07]) et de tester des applications ITS sur route (détection de convoi, prévisibilité de route, suivi de véhicules, *chat...*).

Remerciements

Les auteurs tiennent à remercier Yacine Khaled et Patrick Marlier pour leurs contributions dans la conception et le développement des plates-formes de test. Merci également aux personnes de France Télécom et de l'UTC qui nous ont aidés pour ces tests.

6. Bibliographie

- [ABD 07] ABDALLA G., ABU-RGHEFF M., SENOUCI S., « Current Trends in Vehicular Ad Hoc Networks », in UBIROADS 2007 workshop, IEEE GIIS Marrakech, Morocco, July 2007.
- [BUC 05] BUCCIOL P., MASALA E., KAWAGUCHI N., TAKEDA K., DE MARTIN J., « Performance Evaluation of H.264 Video Streaming over Inter-Vehicular 802.11 Ad Hoc Networks », Proc. of 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), Berlin, Germany, Sep. 2005.
- [CVI] « CVIS making the connection. », http://cvis.odeum.com/download/cvis_first_brochure.pdf.
- [DUC 07a] DUCOURTHIAL B., « About efficiency in wireless communication frameworks on vehicular networks », Invited paper, workshop ACM WIN-ITS co-located with IEEE ACM QShine 2007, Canada, August 2007.
- [DUC 07b] DUCOURTHIAL B., KHALED Y., SHAWKY M., « Conditional transmissions, a strategy for highly dynamic vehicular ad hoc network. », In IEEE WoWMoM'07, 2007.
- [GAS 06] GASS R., SCOTT J., DIOT C., « Measurements of In-Motion 802.11 Networking », In Proc. WMCSA, Apr. 2006.
- [GST] « GST open systems architecture and interface specification. », http://www.gstproject.org/os/documents/DEL_GST_OS_DEV_Reference_Implementation_3_3_2.pdf.
- [HUI 01] HUI F., « Experimental Characterization of Communications in Vehicular Ad Hoc Networks », Thesis Master of Science in Computer Science, University of California Davis, 2001.
- [IPE] « IPERF Traffic Generator. », <http://dast.nlanr.net/Projects/Iperf/>.
- [JER 07] JERBI M., SENOUCI S., MERAIHI R., Y. G.-D., « An improved Vehicular Ad Hoc Routing Protocol for City Environments », IEEE International Conference on Communications (ICC), Glasgow, Scotland, June 2007.
- [KHA 05] KHALED Y., DUCOURTHIAL B., SHAWKY M., « IEEE 802.11 performances for inter-vehicle communication networks », *Vehicular Technology Conference, 2005. VTC 2005-Spring*, vol. 5, 2005, p. 2925-2929.
- [KHA 07a] KHALED Y., DUCOURTHIAL B., SHAWKY M., « A usage oriented taxonomy of routing protocols in VANET », June 2007.
- [KHA 07b] KHALED Y., DUCOURTHIAL B., SHAWKY M., « Conditional transmissions : a communication strategy for highly dynamic vehicular ad hoc networks », *IEEE Transactions on Vehicular Technology, special issue on vehicular communication networks*, vol. 56, n° 6, November 2007, p. 3348-3357.
- [MAD] « Multi-band atheros driver for wifi (madwifi) », <http://madwifi.org/>.
- [OSG] « Wire admin service, OSGi package », org.osgi.service.wireadmin.
- [OTT 04] OTT J., KUTSCHER D., « Drive-thru internet : IEEE 802.11b for automobile users », In IEEE INFOCOM, 2004.
- [PLO] « GNU Plot », <http://www.gnuplot.info/>.
- [SIN 02] SINGH J., BAMBOS N., SRINIVASAN B., D. C., « Wireless lan performance under varied stress conditions in vehicular traffic scenarios. », In *Proceedings. IEEE 56th Vehicular Technology Conference*, vol. 2, 2002, p. 743-747.
- [TOU] « Wireless for linux. », http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/.