Architecture Pour Communication Véhicules-Infrastructure

Bertrand Ducourthial, Farah El Ali

Architecture Pour Communication Véhicules-Infrastructure

Bertrand Ducourthial* — **Farah El Ali***

Laboratoire Heudiasyc
UMR UTC CNRS 6599
Université de Technologie de Compiègne
Centre de Recherches de Royallieu
BP 20529
60205 Compiègne Cedex, France
*{ducourth,elalifar}@utc.fr**

RÉSUMÉ. *Un certain nombre d'applications pour les systèmes de transport intelligents nécessitent des communications véhicules-infrastructures. C'est pourquoi l'orientation aujourd'hui est de lier les réseaux de véhicules à Internet, de manière à offrir une gamme de services élargie, pour le passager, le constructeur automobile ou l'exploitant d'infrastructure. Nous présentons dans cet article une architecture permettant les communications véhicules-infrastructures. Cette architecture permet d'établir une connexion IP v4 ou v6 depuis un véhicule vers l'infrastructure, via la 3G ou les points d'accès WiFi. Cette architecture permet également d'utiliser les protocoles de routage spécifiques aux réseaux de véhicules. Nous utilisons les transmissions conditionnelles afin de remplacer les adresses [DUC 07b]. L'architecture intègre ainsi une découverte de services native afin de déterminer la route vers l'infrastructure. Cette architecture est surtout conçue pour la remontée d'informations depuis les véhicules vers l'infrastructure. Dans cet article, nous présentons l'architecture, sa conception et nos expérimentations.*

ABSTRACT. *A great number of applications for the intelligent transport systems need to be related to the infrastructure. Therefore, researches are more and more oriented to link VANET networks to the Internet in a way to offer a variety of services that can only be offered by Internet to all parties of the ITS process. In this paper, we present an architecture designed for vehicle-infrastructure communications. It allows to establish an IP v4 or v6 connection from the vehicle to the server using 3G networks or WiFi access points. Such an architecture allows to use specific VANET routing protocols; here, we use the conditionnal transmissions in order to replace addresses by conditions [DUC 07b]. This architecture has then a native service discovery in order to find a gateway to the infrastructure without having to go through more manipulations. The architecture presented in this article is mainly conceived to transfer data from vehicles to infrastructure. In this paper, we present the architecture, its components and our experiments.*

MOTS-CLÉS : *ITS, VANET, communications véhicule-infrastructure, IPv4, IPv6, WAVE, 3G.*

KEY WORDS: *ITS, VANET, communications V2I, IPv4, IPv6, WAVE, 3G.*

1. Introduction

Contexte. Dans le cadre des *systèmes de transport intelligents* (en anglais *ITS*), un certain nombre d'applications sont envisagées, telles que l'amélioration de la sûreté, la régulation du trafic routier ou les nouveaux services à bord... L'architecture de ces systèmes est complexe de part la nature des réseaux et des applications envisagés.

Les applications ITS peuvent être classées en quatre familles [DUC 07a]. La première concerne les applications orientées véhicules. Il s'agit de fournir de l'information aux véhicules afin d'adapter leurs automatismes, notamment pour la sécurité routière. La seconde famille concerne les applications orientées conducteur. Le conducteur fera un meilleur usage de la route s'il reçoit des informations sur les dangers à venir, le trafic, etc. La troisième famille concerne les applications orientées passager. La présence de réseaux dédiés initialement à la sûreté laisse à penser que de nouveaux services à bord se développeront (*e.g. infotainment*, accès Internet) [C2C07]. Enfin, la quatrième famille concerne les applications orientées infrastructure. Un meilleur usage des ressources partagées (infrastructures autoroutières) sera possible lorsque les opérateurs publics ou privés obtiendront une information précise de leur état. Il en résulterait des temps de parcours et une émission de CO_2 réduite via des conseils aux automobilistes plus pertinents.

On remarque qu'un certain nombre d'applications nécessitent de remonter de l'information vers l'infrastructure : appel d'urgence, accès Internet, remontée d'information depuis les véhicules pour renseigner les opérateurs sur l'état de la route, etc. Les futurs réseaux de véhicules mêleront donc certainement communications véhicule-véhicule (en anglais *V2V*) et véhicule-infrastructure (en anglais *V2I*). La connexion des véhicules à Internet est donc devenu un sujet de recherche majeur. Mais cet objectif n'est pas simple de par la nature dynamique des réseaux de véhicules, les infrastructures à déployer et le coût limité que devrait représenter l'équipement à bord.

Travaux et projets. Motivés par la sûreté routière et la gestion des infrastructures (réduction des embouteillages), de larges initiatives R&D ont été lancées aux USA (VII, CICAS, IVBSS...), en Europe (CVIS, SafeSPOT, COOPERS, PReVENT, GST, HIGHWAY, FleetNet, SeVeCom, GeoNet...), au Japon (SmartWay, VICS), en Inde (ITSIndia), en Allemagne (NoW), en France (PREDIT)... La plupart d'entre elles inclut des communications V2I. Ainsi l'initiative VII (*Vehicle Infrastructure Integration*) se base sur des communications V2V et V2I pour accroître la sûreté et limiter les embouteillages. Le projet CVIS (*Cooperative Vehicle Infrastructure Systems*) porte également sur la sûreté routière ; il inclut des communications V2V et V2I [CVI]. Le projet Safespot vise à développer un *Safety Margin Assistant* basé entre autre sur les communications V2V et V2I. Le projet PReVENT vise à aider le conducteur à éviter les accidents ou à limiter leur impact ; le sous-projet WILLWARN fait appel aux communications V2V et V2I. Le projet GST (*Global System for Telematics*) porte sur la création d'un standard ouvert pour les services télématiques à bord [GST].

Du point de vue des protocoles réseaux, la mise au point de standard adéquats préoccupe les divers organismes internationaux (IEEE, IETF, ETSI, ISO, SAE, ASTM) ou les consortia d'industriels (OMA, C2C-CC). L'IEEE développe la pile protocolaire WAVE, incluant une extension de la famille de protocoles 802.11 pour les applications ITS. L'ISO développe le standard Calm pour les réseaux de véhicules. L'IETF travaille sur des extensions d'IP (Mobile IP, IPv6, Nemo) et sur l'autoconfiguration dans les réseaux Manet (*Mobile Ad hoc NETwork*) au sein du groupe de travail Autoconf. Le Car-to-Car consortium (C2C-CC) développe et expérimente des protocoles spécifiques aux réseaux

de véhicules. Plus récemment, l'ETSI travaille à l'harmonisation des standards ISO, IETF et IEEE et C2C (ETSI Technical Committee ITS). Ces initiatives sont en cours de développement et donnent lieu à de nombreux travaux ; leur intégration ou inter-opérabilité fait l'objet d'intenses discussions et recherches.

Contribution. Comme nous venons de le voir, les applications ITS et les projets associés motivent la mise au point de protocoles réseaux pour les communications véhicules-infrastructure. Si l'accès à Internet nécessite d'utiliser *in fine* IP, son utilisation native dans les communications véhicule-véhicule fait débat. Une telle uniformisation des réseaux a des avantages mais aussi des inconvénients en terme de contrôle ajouté (*overhead*), et pose de réels problèmes pour l'auto-configuration des adresses [CAL 09].

Nous proposons dans cet article une architecture dédiée à la remontée d'information depuis les véhicules vers l'infrastructure, utilisant soit des points d'accès WiFi, soit des connexions 3G. Les données à remonter sont acheminées de véhicule en véhicule jusqu'à trouver une passerelle vers l'infrastructure. Notre architecture supporte IPv4 et IPv6 pour atteindre le serveur Internet final, selon les disponibilités du réseau d'infrastructure trouvé. Cependant elle ne repose pas sur IP pour les communications véhicule-véhicule. Nous utilisons un protocole adapté – les transmissions conditionnelles [DUC 07b] – qui inclut naturellement la découverte de la passerelle vers l'infrastructure, et qui évite tout surcoût de contrôle en cas d'absence d'émission vers Internet.

Dans la section 2, nous présentons la problématique de l'accès Internet depuis les véhicules et résumons les principales contributions dans ce domaine. Nous détaillons le scénario envisagé et notre architecture dans la section 3. Les sections 4, 5 et 6 présentent les composants de notre architecture. La section 7 résume nos expérimentations. Nous concluons en section 8.

2. Problématique de l'accès à Internet depuis les véhicules

Dans cette section, nous résumons les solutions envisagées pour accéder à Internet depuis les véhicules. On distingue quatre acteurs principaux dans ce domaine : l'IEEE, l'ISO, l'IETF et le C2C-CC.

Tout d'abord l'IEEE a étendu sa famille de protocoles 802.11 en ajoutant le 802.11p, s'inspirant pour cela du standard ASTM E2213-03, lui-même basé sur le 802.11a. Ce protocole modifie couche physique et couche MAC pour s'adapter aux réseaux de véhicules, en conformité avec la bande DSRC (*Dedicated Short Range Communication*). En complément, l'IEEE a défini la famille de protocoles 1609, dite *WAVE*[1] (*Wireless Access in Vehicular Environments*), pour l'accès sans fil dans les réseaux de véhicules [IEE04]. Ce standard, structuré en quatre composantes (1609.1 à 1609.4), définit l'architecture, le modèle de communication, la structure de management, la sûreté et l'accès physique. Au final, 802.11p et WAVE spécifient une pile protocolaire complète. Le standard 1609.3 inclut le protocole WSMP (*Wave Short Messages Protocols*) pour les communications inter-véhicules, présenté comme une alternative à IPv6 [RES]. Dans ce protocole, les messages sont

1. Le terme DSRC désignait des concepts différents depuis la gamme de fréquences jusqu'aux types d'applications. Le terme WAVE proposé par l'IEEE devait clarifier les usages du terme DSRC en les limitant [IEE04]. Actuellement, la « bande DSRC » ne désigne pas les mêmes gammes de fréquences d'un continent à l'autre.

routés avec un *application class identifier* (ACID) et un *application context mark* (ACM) en lieu et place de l'adresse IP et du numéro de port [IEE 07].

Les développements de l'IEEE se sont fait en lien avec l'ISO, plus particulièrement le « *Technical Comittee* 204 *Intelligent Transport Systems*, *Working Group* 16, *Wide Area Communications* » en charge des communications moyennes et longues portées, qui travaille sur le standard Calm [2] *Continuous Air-Interface for Long and Medium range telecommunication* [Cal]. Il s'agit en fait plus d'un référentiel que d'un protocole, dont le but est de standardiser les communications par commutation de paquets dans les réseaux hétérogènes en environnement mobile [EVE 06]. Le but de Calm est d'offrir des communications en continue de manière transparente à l'utilisateur à travers des réseaux et des interfaces de communication variées, telles que 802.11, 802.11p, 802.15, 802.16e, 802.20, réseaux cellulaires 2G/3G/4G et systèmes ITS nationaux. Calm intègre à la fois les travaux de l'IEEE et de l'IETF.

De son côté, l'IETF travaille depuis quelques années sur les réseaux mobiles, les réseaux ad hoc, et plus récemment les réseaux de véhicules. La problématique adressée est celle d'un déploiement complet d'IP, en donnant à chaque véhicule une adresse. Ces travaux portent essentiellement sur IPv6.

Le protocole Mobile IPv6 repose sur le principe de mise à jour d'une adresse dite *care_of address* ou adresse temporaire. Le mobile dispose alors de deux adresses, l'une permanente liée à son réseau d'origine et l'autre temporaire liée au réseau visité. Le mobile met à jour son adresse temporaire à chaque fois qu'il change de réseau, et envoie cette adresse au *home agent* de son réseau d'origine pour qu'il l'enregistre. Dès lors, les messages arrivant au *home agent* et destinés au mobile qui est en mouvement, seront redirigés vers ce mobile, qui informera alors l'émetteur de sa nouvelle adresse. La communication s'enchaîne ensuite sans l'intervention du *home agent*. On remarque ici que, contrairement à Mobile IPv4, il y a optimisation du routage car les paquets n'ont pas forcément à passer par le *home agent*.

Le protocole Mobile IPv6 ne supporte pas la mobilité des réseaux, composés de plusieurs adresses alors que certains travaux envisagent d'associer une adresse IP à chaque ordinateur embarqué au sein d'un même véhicule. Le protocole Nemo Basic Support (RFC 3963) gère cette problématique tout en étant basé sur le modèle Mobile IPv6 tandis que Nemo Extended Support étudie les problématiques d'optimisation de multidomociliation et d'optimisation de routage, sans reposer par contre sur Mobile IPv6 [DEV 05, ERN]. Nemo BS ne modifie pas les *Mobile Network Nodes* (MNN) situés derrière le *Mobile Router* (MR). Le déplacement du MR n'entraîne pas de changement de point d'ancrage pour les MNN : seul le MR change d'adresse extérieure. Le *home agent* du MR encapsule alors tous les paquets dont le préfixe de l'adresse de destination correspond au préfixe du réseau mobile.

La problématique de l'assignement d'une adresse IP à un véhicule est adressée dans le *Ad Hoc Network Autoconfiguration (Autoconf) Working Group*. La nature ad hoc multisauts du réseau de véhicules empêche d'utiliser les protocoles d'auto-configuration d'adresses tels que ceux des RFC 4861 et 4862. Il n'y a pas à l'heure actuelle de standard pour assigner des adresses IP aux

2. Depuis 2007, Calm signifie *Communication Architecture for Land Mobile* (auparavant *Continuous Air-Interface for Long and Medium range telecommunication*).

véhicules [CAL 09], ni énormément de travaux publiés. Nous en résumons deux. Dans [FAZ 07], la topologie des réseaux Vanet est supposée composée de petits convois linéaires indépendants. Des *leaders* sont choisis parmi les véhicules ; ils agissent en tant que serveurs DHCP. Cette solution de type *distributed DHCP* garantit l'unicité de l'adresse au sein de chaque petit convoi mais deux véhicules distants pourraient avoir la même adresse.

La solution proposée dans [BAL 08] se base sur l'architecture C2C-CC et sur la technique SLAAC (*Stateless Address Autoconfiguration*), qui repose sur une signalisation NDP (*Neighbour Discovery Protocol*) pour vérifier l'unicité des adresses IPv6 (ce qui suppose que tout nœud du réseau peut communiquer avec les autres). GeoSAC étend SLAAC aux réseaux géographiquement distribués en utilisant le protocole de routage géographique du C2C-CC, qui permet d'offrir une zone de *broadcast* limitée, facilitant la configuration des adresses.

Le consortium C2C-CC promeut le développement des réseaux de véhicules. Il a développé une pile protocolaire complète spécifique [C2C07], mais elle peut aussi intégrer IPv6, Mobile IP et Nemo. Les couches physiques envisagées sont 802.11p, le WiFi et la 3G.

3. Scénarios envisagés et architecture proposée

Scénario. Notre architecture a pour but de permettre aux applications à bord des véhicules d'envoyer des données vers un serveur de l'infrastructure. Il s'agit donc de communications véhicule-vers-infrastructure, qui ne donnent pas lieu à un retour depuis le serveur vers le véhicule émetteur. Même si des extensions sont envisageables, nous ne les aborderons pas ici. Les applications visées concernent essentiellement la remontée d'informations produites à bord des véhicules par les capteurs et calculateurs embarqués, tels que : position, vitesse, adhérence, luminosité, pluie, densité du voisinage, etc. Ces informations (et bien d'autres) sont produites par des capteurs à bord. Elles permettent d'évaluer les conditions de circulation, d'informer de l'état d'un camion ou de gérer une flotte de véhicules (*e.g.* société de livraison ou de dépannage...). À moins d'une exploitation nominative (*e.g.* position d'un véhicule donné, appartenant à une flotte) l'information devra être consolidée au départ (*e.g.* moyennes des vitesses alentours...) et/ou à l'arrivée afin d'être exploitable, mais cette problématique n'est pas adressée dans cet article, qui se focalise sur l'architecture réseau.

En fin de communication, les informations sont collectées par un serveur web, de sorte que la communication utilisera HTTP sur TCP sur IP (IPv4 ou IPv6 selon les disponibilités du réseau d'infrastructure). Les réceptions seront gérées par une page web dynamique écrite en PHP. La connexion TCP sera établie entre un seul véhicule, dit *véhicule passerelle*, et le serveur. Le véhicule passerelle n'est pas forcément celui qui a produit l'information.

Architecture. Une application embarquée désirant émettre des données (application APP sur la figure 1) vers le serveur web contactera sa passerelle locale. Si celle-ci dispose d'un accès WiFi ou 3G, alors elle émettra immédiatement les données vers Internet. Sinon, cela dépend du paramètre d'urgence des données. Si l'urgence est basse, alors la passerelle locale attend un certain temps en espérant trouver un point d'accès WiFi. Si l'urgence est forte (ou que le délais d'attente est atteint), alors la passerelle locale diffusera le message à transmettre aux véhicules proches. Si l'un d'eux possède une passerelle capable d'émettre vers Internet, alors il émettra le message. Dans le cas contraire, le message sera retransmis de proche en proche jusqu'à trouver une passerelle.

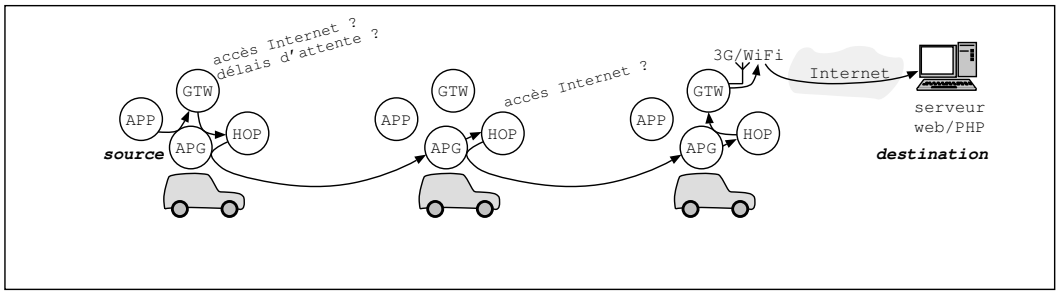

Figure 1. *Architecture de communication véhicule-infrastructure. APG : airplug (gère les communications intra et inter-véhicules), APP : application générant les données, HOP : multi-sauts VANET (transmissions conditionnelles), GTW : passerelle ayant ou non un accès vers Internet.*

Notons qu'il est possible qu'aucune passerelle ne soit trouvée dans un temps ou délais raisonnable. Dans ce cas, le message arrêterait d'être retransmis. Dans certains cas défavorables, le message pourrait donc ne pas atteindre le serveur. Cependant, vu les applications visées, un message trop vieux n'a plus grand intérêt et il serait préférable d'en envoyer un plus récent (contenant des informations produites plus récemment par les capteurs et calculateurs embarqués). À l'inverse, il est possible dans certains cas défavorables que le serveur reçoive plusieurs fois le même message. Cette situation n'a d'inconvénient que la surcharge du réseau et du serveur qu'elle entraîne. L'équilibre entre la probabilité pour un message de ne pas atteindre le serveur, et celle de l'atteindre plusieurs fois dépend du paramétrage de notre architecture. Ce paramétrage est lié à l'inconvénient d'une surcharge des ressources comparé à celui d'une perte de données, sachant qu'elles sont émises régulièrement par les capteurs. Tout dépend donc de l'importance des données remontées. Ce paramétrage devrait évoluer en fonction de la densité de véhicules, du taux d'équipement des routes en point d'accès WiFi et de celui des véhicules en communication 3G. Remarquons que ce dilemme entre ressources réseaux et garantie de service est assez classique.

Les communications de véhicules à véhicules ne nécessitent pas d'utiliser IP. Seul le véhicule passerelle (qui émettra effectivement les données vers Internet) réalise une connexion TCP/IP. De ce fait, tout protocole de routage spécifique aux réseaux de véhicules peut être employé. Nous utilisons les transmissions conditionnelles [DUC 07b], dont le but est de substituer aux adresses des conditions déterminant si le message reçu doit être retransmis aux couches applicatives et/ou aux véhicules voisins. Cette technique de transmission, qui pourrait, sous certains aspects, se comparer à un routage de type *content-based*, permet d'éviter de rechercher les adresses des destinataires et des relais. En outre, elle inclut par nature une recherche de passerelle. Les conditions sont évaluées à la réception, évitant ainsi tout message de contrôle pour connaître le voisinage (et au delà), ce qui est coûteux et parfois vain dans un réseau dynamique tel qu'un réseau de véhicules. Notre architecture limite donc au maximum le contrôle dans le réseau. Si aucun message n'est à transmettre, alors très peu de messages de contrôle sont émis. Ne subsistent que ceux nécessaires à la découverte des *hot-spot* WiFi.

Autre avantage, notre architecture garantit le respect de la vie privée (*privacy*) [FON 07]. L'un des freins au déploiement de certaines applications ITS concerne le respect de la vie privée des

automobilistes, alors qu'on envisage des GPS à bord, des cartes 3G et des remontées de données concernant leur vitesse, leurs trajets, etc. Ici, le fait que les données n'émanent pas forcément du véhicule passerelle protège et l'émetteur et la voiture passerelle (impossible de distinguer les données provenant du véhicule passerelle lui-même ou d'un autre véhicule). La connexion IP de la passerelle au serveur et non de l'émetteur au serveur agit comme un proxy pour une connexion web : s'il n'interdit pas l'authentification, il permet aussi de l'éviter.

Seul le véhicule passerelle (qui lance la requête vers le serveur web) nécessite une adresse IP. Mais puisqu'il est à proximité de l'infrastructure et qu'une seule requête est envoyée à la fois, il n'y a pas à proprement parlé de problématique d'auto-configuration d'adresses ni de gestion de mobilité.

Composants. La réalisation de l'architecture est décrite dans les sections suivantes, et illustrée sur la figure 1. Nous utilisons l'intergiciel de communications Airplug (abrégé APG sur la figure), dédiés aux réseaux dynamiques tels que les réseaux de véhicules (section 4). Airplug permet de simplement développer de nouveaux protocoles en espace utilisateur (*user-space*). APP désigne une application générant les données à transmettre au serveur Internet. Pour gérer les communications multi-sauts dans le réseau de véhicules, nous utilisons les transmissions conditionnelles (section 5) développées sous la forme d'une application Airplug appelée HOP [DUC 07b]. L'application passerelle appelée GTW effectue l'émission vers Internet lorsqu'elle dispose d'un accès WiFi ou d'une carte 3G embarquée et sinon elle confie le message à HOP qui recherche une passerelle vers Internet.

4. Airplug : intergiciel de communication léger pour réseaux dynamiques

Bien que d'autres implémentations soient envisageables, nous avons basé la réalisation de notre architecture sur Airplug, que nous décrivons dans cette section. Airplug est un intergiciel de communication léger pour réseaux dynamiques [DUC 07a, DUC 09]. Il se caractérise par sa robustesse et sa simplicité dans l'organisation des échanges de messages intra- et inter-véhicules, ce qui est bien adapté à la dynamique des réseaux.

L'architecture de Airplug repose sur les facilités procurées par les systèmes d'exploitations standards : allocation de ressources, ordonnancement des processus, gestion du « temps-réel », etc. Cela évite toute redondance entre l'intergiciel et le système d'exploitation (qui doit être POSIX). Airplug étant développé en « espace utilisateur », aucune modification n'est faite au niveau du noyau, ce qui accroît sa portabilité. Le programme-cœur Airplug (abrégé APG sur la figure 1) s'intercale entre les interfaces réseaux et les applications, ce qui permet de se prémunir contre une application qui consommerait trop de ressource (*e.g.* applications tiers boguées).

Toutes les communications via Airplug se font par passage de messages. Un message provenant d'une application peut être envoyé à plusieurs applications, locales ou distantes. Cependant, une application ne peut pas recevoir les messages adressés à toutes les applications, ni émis par une application non locale sans s'être abonnée auprès d'Airplug à l'application émettrice. Ce principe d'abonnement (confiance relative en local, confiance limitée à distance) permet à une application de contrôler ses réceptions. En outre, cela augmente la robustesse de l'architecture en évitant les problèmes en cascades dans le cas d'applications boguées (sur lesquelles il est plus facile d'agir lorsqu'elles sont locales que lorsqu'elles sont distantes). Les messages utilisent un format d'adressage spécifique. La destination d'un message est composée de deux champs : un champ *zone* qui

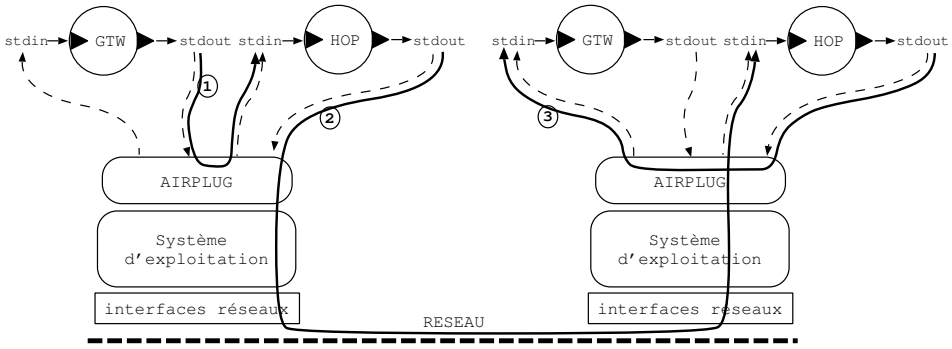

Figure 2. *Architecture de Airplug, communications intra-véhicules (1) et (3), inter-véhicules (2).*

contient la zone d'envoi du message suivi du nom de l'application destinataire. Cet adressage simplifié permet de supporter la dynamique du réseau et s'avère néanmoins suffisante pour construire protocoles de communication et applications réparties. La zone peut être interne (LCH pour *localhost*) ou externe (AIR), c'est-à-dire composée des voitures dans le voisinage, ou bien les deux (ALL). Mais elle peut également être plus spécifique (nom ou adresse d'une voiture avoisinante).

La réalisation des communications utilisées par Airplug se fait de la manière la plus simple et robuste possible : en utilisant les entrées et sorties standard. Cela garantit une indépendance complète du langage de programmation utilisé pour développer les applications. Pour chaque processus lancé par Airplug, l'entrée et la sortie standard sont redirigées depuis et vers Airplug via un *pipe* (figure 2). Ainsi, chaque fois qu'un processus écrit sur sa sortie standard, Airplug reçoit les données via ce lien, et chaque fois que Airplug écrit sur un de ces liens, le processus correspondant pourra lire les données sur son entrée standard. Comme les interfaces réseaux sont aussi gérées par Airplug, les applications accèdent au réseau de la même manière qu'elles le feraient pour communiquer avec d'autres applications locales, simplement en écrivant sur leur sortie standard. Airplug reçoit les données envoyées par les processus et les envoie vers l'interface désirée.

Avec Airplug, le développement de nouveaux protocoles de communication est réalisé en espace utilisateur, au sein d'un processus qui recevra les données à émettre via son entrée standard et qui se chargera de les transmettre en les donnant à Airplug via sa sortie standard. Il est possible de cascader plusieurs protocoles (*e.g.* routage, transport, etc.) sur ce modèle. Le prototypage de nouveaux protocoles est facilité, de même que les solutions inter-couches (*cross-layering*). Airplug peut court-circuiter les piles protocolaires du système d'exploitation en utilisant des sockets raw. La figure 2 détaille les relations entre l'application passerelle GTW et le protocole HOP au sein de notre architecture V2I : GTW émet localement vers l'instance locale de HOP (1), qui émet vers l'instance distante de HOP (2), qui transmet à l'instance distante GTW (3).

5. Transmissions conditionnelles : routage adapté aux réseaux dynamiques

Les transmissions conditionnelles s'apparentent à un routage dans lequel des conditions logiques ont été substituées aux adresses [DUC 07b]. Un message envoyé par le module de retransmissions conditionnelles est accompagné de deux conditions, CUP et CFW (figure 3). À la réception, si CUP est vraie, le message est transmis à la couche applicative. Si CFW est vraie, le message est retransmis aux véhicules proches. Toutes sortes de conditions logiques peuvent être utilisées (y compris des conditions testant d'éventuelles adresses IP ou géographiques). Mais les conditions se révélant les plus intéressantes sont celles portant sur la distance, la durée, la corrélation de trajectoire (permettant de déterminer si la voiture qui reçoit suit ou non l'émetteur).

En évaluant dynamiquement les conditions à la réception, le protocole s'accommode mieux de la dynamique que ceux qui reposent sur des adresses (y compris géographiques). En effet, une adresse représente un nom de machine et/ou une position dans le réseau. Déterminer le destinataire immédiat du prochain saut côté émetteur implique connaître le voisinage, ce qui nécessite des messages de contrôle. Or, le contrôle nécessaire augmente avec la dynamique tandis que son efficacité diminue car l'information collectée est rapidement périmée. De même, déterminer une position dans un réseau est complexe lorsque la dynamique est grande, que le positionnement soit logique, hiérarchique ou géographique. Les protocoles de type geocast sont généralement contraints d'augmenter la zone de recherche du destinataire pour prendre en compte ses éventuels déplacements, ce qui impact d'autant plus de véhicules que la dynamique est grande.

Les transmissions conditionnelles ont été implémentées sous la forme d'une application compatible Airplug dénommée HOP, et testée sur route [DUC 09]. Pour les besoins de notre architecture, nous avons complété cette application afin qu'elle accepte des messages particuliers, qui lui précisent certains mots-clés à considérer comme vrais lors de l'évaluation des conditions. Ces messages ne sont acceptés que s'ils émanent des applications locales au véhicule. Ainsi, l'application GTW (présente sur chaque véhicule) envoie périodiquement de tels message à HOP afin de lui indiquer si elle a détecté un *hot spot* WiFi (mot-clé wifi) ou si elle possède une carte 3G (mot-clé 3G).

Lorsqu'une application GTW ne peut émettre directement le message vers Internet (absence de carte 3G et de point d'accès WiFi) et que, de par l'urgence du message, elle ne peut attendre davantage la proximité d'un *hot spot* WiFi, elle le transmet à HOP accompagné de deux conditions (CUP et CFW). Le HOP initiateur enverra alors le message de GTW accompagné des deux conditions fournies et d'informations complémentaires nécessaires à l'évaluation des conditions (figure 3). La condition CUP «wifi $\vee$ 3G» permettra de transmettre le message aux applications GTW qui possèdent effectivement un point d'accès Internet. La condition CFW « $\neg$wifi $\wedge$ $\neg$3G $\wedge$ dst<2000 $\wedge$ dur<180» permettra par exemple de retransmettre le message si localement aucun accès Internet n'a été détecté et si la distance parcourue est inférieure à 2km et si la durée est inférieure à 3 minutes. Dans ce cas, les informations complémentaires seront la date et la position de la voiture source au moment de l'envoi (obtenues via le GPS embarqué), de manière à ce que chaque relais potentiel puisse calculer sa distance à l'émetteur et l'ancienneté du message reçu. Une estampille empêche tout traitement d'un message déjà reçu.

Il est important de remarquer que l'utilisation de HOP et de telles conditions réalise une découverte de service induite puisqu'il n'est pas nécessaire d'ajouter un pré-traitement pour rechercher un accès vers Internet.

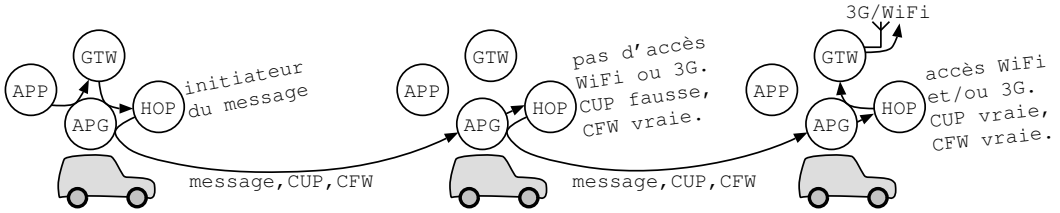

Figure 3. *Transmissions conditionnelles : un message est accompagné des conditions CUP et CFW.*

6. Passerelle : découverte de réseaux, émission ou retransmission

En complément des nouvelles fonctionnalités ajoutées à HOP, une nouvelle application Airplug a été développée pour les besoins de notre architecture V2I. Cette application passerelle, dénommée GTW, a pour rôle de faire le lien entre le réseau de véhicules et le réseau Internet.

GTW vérifie périodiquement la disponibilité des interfaces vers les réseaux externes. Les interfaces utilisées seront un sous-ensemble des interfaces détectées, selon le paramétrage manuel ou automatique. Il est en effet possible de restreindre le choix à la 3G, aux points d'accès WiFi, au LAN (pour les tests sur table, voir section suivante), à IPv4 ou IPv6. GTW signale périodiquement à HOP les réseaux disponibles via une communication intra-véhicule lui indiquant les mots-clés à évaluer à vrais lors de l'examen des conditions associées aux messages reçus (cf. section précédente).

Dès lors qu'une application souhaite émettre des données vers un serveur Internet, elle les transmet à l'instance locale de GTW, présente sur son véhicule avec une information d'urgence. Si cette instance dispose d'une connexion vers Internet, elle émet les données immédiatement. Sinon, si l'urgence est basse, elle attend de découvrir une connexion. Dans le cas contraire ou lorsque le délais a expiré, elle les transmet à HOP qui se chargera de trouver une passerelle grâce à la découverte de service induite par la technique des retransmissions conditionnelles.

L'application GTW a été développée en Tcl/Tk (le choix initial de Tcl/Tk pour les applications Airplug s'explique par le fait qu'il nous est plus facile de les adapter à Network Simulator par la suite). Nous n'avons pas trouvé d'implémentation d'IPv6 fonctionnelle en Tcl. Aussi avons-nous développé un lanceur d'application Tcl/Tk en C dénommé `launchtk` permettant d'ajouter des fonctionnalités au Tcl grâce à la libtcl. Nous avons ainsi développé une fonction C `send_IPv6` utilisable depuis les scripts Tcl. L'application GTW est lancée via launchtk, lui-même lancé par Airplug (fork + exec et redirection des pipes, cf. section 4).

7. Validation expérimentale

Pour valider l'ensemble des développements et, par là, l'architecture que nous avions imaginée, nous avons utilisé 4 véhicules embarquant des PC (Dell mini-9 Modèle DP118) sous Ubuntu (v8.04 Hardy Heron). Tous sont équipés d'un GPS (nécessaire à l'évaluation des conditions fournies à HOP) et d'une carte WiFi externe à connectique USB (Alfa AWUS036EH) permettant de connecter une antenne sur le toit des véhicules (D-link). Un seul PC (voiture passerelle) est équipé d'une carte 3G

Figure 4. *Plate-forme expérimentale composée de mini Dell sous Linux, de cartes WiFi externes, d'antennes externes et de GPS. Le Dell de droite dispose d'une carte 3G (connectiques USB).*

(HUAWEI E510). Le serveur Internet est le serveur web Apache du laboratoire, auquel nous avons ajouté une page web spécifique en PHP, qui stocke les données dans un fichier. La première voiture émet des messages, qui seront relayés (grâce à HOP) dans le convoi jusqu'à atteindre la voiture passerelle en fin de convoi. Cette dernière relaie le message vers le serveur du laboratoire via la 3G (figure 3). Le délai moyen d'un saut véhicule-véhicule est d'environ 30 ms tandis que le délai moyen entre la voiture passerelle et le serveur varie entre 250 et 350 ms. Le taux de perte global est d'environ 20%.

Nos expériences nous ont permis de vérifier le bon fonctionnement des échanges inter-applications impliqués dans l'architecture : APP–GTW, GTW–HOP, GTW–Internet (les communications HOP–HOP avaient déjà été validées). Les tests avec IPv6 n'ont pour l'instant été effectués qu'avec un LAN privé et un serveur web spécifique faute de disposer d'adresses IPv6 publiques. HOP acceptant maintenant de considérer tout mot-clé comme étant vrai dans les expressions, il nous a été facile d'ajouter le mot-clé LAN pour ce test. Il ressort de nos expérimentations que l'architecture est pleinement opérationnelle et permet de remonter des données depuis les véhicules jusqu'au serveur du laboratoire, en passant par le réseau WiFi ou la 3G, en IPv4 ou en IPv6, selon disponibilité.

8. Conclusion

Dans cet article, nous avons présenté la problématique de l'accès Internet pour les réseaux de véhicules. Nous nous sommes intéressés à la remontée d'informations produites par les véhicules jusqu'à un serveur de l'infrastructure. Nous avons proposé une architecture permettant de rechercher une passerelle vers Internet dans le réseau de véhicules, exploitant un routage spécifique dans le réseau Vanet, et une connexion HTTP/TCP/IP depuis le véhicule passerelle jusqu'au serveur web. Nous avons présenté les avantages d'une telle architecture en terme de paramétrage de l'encombrement réseau, de limitation du surcoût lié au contrôle, de la découverte de passerelle induite, de respect de la *privacy*. Puis nous avons détaillé les éléments constitutifs de l'architecture : l'intergiciel Airplug, les transmissions conditionnelles, l'application passerelle.

Les expérimentations menées ont validé l'intérêt de cette architecture, qui a été couplée à une application répartie de collecte de données, permettant de remonter sur un serveur de l'infrastructure

par exemple la vitesse moyenne d'un convoi et ce, sans qu'il soit nécessaire d'équiper tous les véhicules d'un accès 3G. Notre contribution s'est focalisée sur la remontée d'information. Nos futurs travaux concerneront de nouveaux protocoles de transport, adaptés aux réseaux de véhicules qui permettraient d'obtenir des scénarios plus variés et une communication bidirectionnelle entre les voitures et les serveurs d'infrastructure, à l'instant de ce qui est envisagé dans les standards en cours de définition.

Remerciements. Nous voudrions remercier MM. Anthony Buisset, Thierry Ernst et Manabu Tsukada pour leur aide, conseil ou expertise.

9. Bibliographie

[BAL 08] BALDESSARI R., BERNARDOS C., CALDERON M., « GeoSAC-scalabe address autoconfiguration for VANET using geographic networkin concepts », *Proc. IEEE PIMRC 2008, Cannes, France*.

[C2C07] « CAR 2 CAR Communication Consortium Manifesto, Overview of the C2C-CC System », `http://www.car-2-car.org/fileadmin/downloads/C2C-CC_manifesto_v1.1.pdf`, August 2007.

[Cal] « CALM, Continuous Communications for Vehicles », `http://www.calm.hu/`.

[CAL 09] CALDERON M., MOUSTAFA H., BERNARDOS C., BALDESSARI R., « *IP Address autoconfiguration in vehicular networks*, chap. 9 in *Vehicular Networks, Techn., Standards and App.*, Auerbach, 2009.

[CVI] The CVIS project, http ://www.cvisproject.org/.

[DEV 05] DEVARAPALLI V., WAKIKAWA R., PETRESCU A. THUBERT P., « Network Mobility (NEMO) Basic Support Protocol », IETF RFC 3963, January 2005.

[DUC 07a] DUCOURTHIAL B., « About efficiency in wireless communication frameworks on vehicular networks », *Proceeding of the ACM WIN-ITS workshop (with IEEE ACM QShine'07)*, 2007.

[DUC 07b] DUCOURTHIAL B., KHALED Y., SHAWKY M., « Conditional transmissions : performances study of a new communication strategy in VANET », *IEEE TVT*, vol. 56, n° 6, November 2007, p. 3348 – 3357.

[DUC 09] DUCOURTHIAL B., KHALFALLAH S., « A platform for road experiments », *Proc. of the 69th IEEE VTC2009-Spring*, Barcelona.

[ERN] ERNST T., « Le support des réseaux mobiles dans IPv6 ».

[EVE 06] EVENSEN K., « CALM, Continuous communications for vehicles », November 2006.

[FAZ 07] FAZIO M., PALAZZI P., DAS S., GERLA M., « Facilitating real-time applications in VANETs through fast address auto-configuration », *Proc. of IEEE CCNC/NIME*, Las Vegas, January 2007.

[FON 07] FONSECA E., FESTAG A., BALDESSARI R., AGUIAR R., « Support of anonymity in VANETs, putting pseudonymity into practice », *Proc. IEEE WCNC*, Hong-Kong, March 2007.

[GST] The GST project, http ://www.gstforum.org/.

[IEE 07] IEEE, « Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Architecture », March 2007.

[IEE04] « Amendment to Standard [for] Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - specific requirements - Part II : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications : Wireless Access in Vehicular Environments », `http://www.ieee802.org/secmail/doc00268.doc`, 05 2004,

[RES] RITA/ITS, « IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE) », `http://www.standards.its.dot.gov/fact_sheet.asp?f=80`.