République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ammar Telidji Laghouat

Faculté des Sciences

Département Mathématiques et Informatique

Ecole Doctorale IRM

Informatique Répartie et Mobile

Mémoire

En vue de l'obtention du diplôme de

Magistère en Informatique

Présenté par:

Benarfa Abdelmadjid

Thème:

Optimisation des performances des protocoles de transport dans les VANets par une approche cross-layer

Soutenu publiquement devant le jury composé de:

Mr	Ouinten Youcef	Président	Maître de conférences A	UATL
Mr	Yagoubi Mohamed Bachir	Examinateur	Professeur	UATL
Mme	Cherroun Hadda	Examinatrice	Maître de conférences A	UATL
Mr.	Lagraa Nasreddine	Rapporteur	Maître de conférences A	UATL

Année Universitaire 2013/2014

À mes parents,

....

À mes frères et sœurs,

....

À mes amis,

....

À tous les professeurs,

....

Je dédie ce modeste travail

Benarfa Abdelmadjid

Remerciements

Je tiens à remercier en premier lieu mon Dieu qui m'a donné la santé, la force physique et intellectuelle et le courage pour mener à bien la réalisation de ce travail.

Je remercie chaleureusement mes parents qui m'ont beaucoup aidé par leurs conseils, leurs encouragements depuis mon enfance jusqu'à ce que je suis arrivé à ce point-là, sans oublier aussi mes sœurs, mes frères et tous les membres de ma famille.

Faire un magister c'est apprendre à devenir chercheur. C'est un chemin long et semé d'embuches qui ne peut se réaliser sans un soutien important.

C'est pourquoi je voudrais remercier mon directeur de thèse ***M. Nasreddine LAGRAA*** *pour son encadrement, les réunions de travail et les discussions tant scientifiques que personnelles que nous avons pu avoir, je lui adresse donc un grand merci.*

Merci à tous ceux qui ont participé de près ou de loin à l'aboutissement de ce travail.

Benarfa Abdelmadjid

Résumé

Actuellement, les réseaux véhiculaires ou VANETs ont attirés beaucoup d'attention dans la communauté de recherche et dans les industries (automobiles, télécommunications,... etc).

La particularité des VANETs provient des communications qui peuvent être établies pour offrir une diversité d'applications aux usagers de la route.

Les performances des applications de haut niveau (multimédia, jeux, Navigation sur Internet,... etc.) sont liées essentiellement aux protocoles de la couche transport utilisés dans un réseau très dynamique.

À cause de cette forte mobilité et la nature des liens radio qui ne sont pas toujours fiables, les données transmises ou reçues par les véhicules peuvent être perdues pour plusieurs raisons (interférences, déconnexion,...).

Dans ce mémoire, on a dressé en premier lieu un bilan sur les travaux proposés dans le cadre de l'amélioration de performances des protocoles TCP dans les différents types de réseaux, afin d'identifier leurs carences et les points qu'il faut encore améliorer ou ceux qui ne sont pas encore traités.

Par la suite on a conçu un protocole cross-layer qui augmente les performances et répond aux exigences de tels réseaux.

Mots-clés : VANets, TCP, Congestion, Transport, communication inter-couches.

Abstract

Currently, vehicular networks VANET have attracted more attention in both the research and industries communities (automobiles, telecommunications, ... etc).

The characteristics of VANETs come from communications that can be established to provide a variety of applications for the road users.

The performance of high-level applications (multimedia, gaming, Internet browsing, ... etc) is mainly related to the transport layer protocol used in the network characterized by high mobility.

Due to this high mobility, and the nature of radio links that are not always reliable, the data transmitted or received by vehicles may be lost for several reasons (interference, disconnection, ...).

In this thesis, we try first to take a global overview about the proposed protocols in the context of improving TCP performance in different types of networks in order to identify their weaknesses and points to be improved or even those which are not yet processed.

Secondly we've tried to design a cross-layer protocol that improves performance and meets the requirements of such networks.

Key-words : VANETs, TCP, wireless environment, congestion, cross-layer design.

Table des matières

Liste des figures

Introduction générale

Contexte

Depuis plusieurs années, les systèmes de transport intelligents (ITS) sont considérés comme un sujet d'actualité qui attire de plus en plus l'attention de la communauté de recherche et celle des industries (automobiles, télécommunications ... etc.). Dans tels systèmes, les réseaux de communications inter-véhiculaires sont devenus une composante essentielle pour la mise en œuvre et l'amélioration des performances globales en termes de sécurité ou de confort.

Ainsi, plusieurs applications sont développées dans le but d'améliorer la sécurité des individus, de réduire les congestions et les embouteillages ou de limiter l'impact des véhicules sur l'environnement,... etc ; ces applications ont généralement des contraintes différentes, les applications de sécurité nécessitent par exemple un délai très court.

Tant que les applications visant à augmenter le confort des passagers (Accès à Internet, téléchargement des fichiers ou jeu en réseau,... etc) ont des exigences sur la fiabilité. La majorité des deux types d'applications sont généralement implémentées dans les couches supérieures ; la couche dite « facilities » ou la couche application, et elles ne peuvent pas être mises en œuvre sans l'utilisation d'un protocole fiable de la couche transport.

Problématique

La plupart de ces applications utilisent le protocole TCP au niveau de la couche transport, cependant des travaux de recherche ont montré que ce protocole ne peut pas être directement appliqué aux réseaux sans fil en raison de la mobilité des nœuds et l'incertitude des liens qui varient dans le temps. Ces caractéristiques produisent des erreurs de transmission fréquentes dans les connexions TCP.

Par exemple, si un expéditeur TCP ne reçoit pas des acquittements du récepteur qui quitte la portée de l'expéditeur, les événements d'expiration du délai d'attente de transmission des segments se déclenchent, et dans ce cas, TCP interprète ces événements comme congestion du réseau, en conséquence il exécute un contrôle de congestion dans des situations inappropriées.

Donc, tous les protocoles de transport proposés pour les réseaux filaires ne sont pas adaptés aux réseaux sans fil ou ad hoc. En plus, les protocoles proposés pour les réseaux mobiles comme Indirect TCP (ITCP)[4], WESTWOOD [5], TCP-ELFN [6], Ad-Hoc TCP (ATCP)[3] ne peuvent pas être appliqués aux réseaux véhiculaires à cause des caractéristiques spécifiques de ces réseaux telle que la forte mobilité.

A travers toute la littérature que nous avions pu trouver, très peu de protocoles de transport sont développés pour les réseaux VANet, exactement trois protocoles ; VanetTCP [7], MCTP (Mobile Control Transport Protocol) [8], et VTP (Vanet Transport Control Protocol) [9]. Ces protocoles ont essayé de résoudre le problème de l'adaptation avec la forte mobilité. Mais, malheureusement les mécanismes de contrôle de perte souffrent de l'inefficacité dans certains cas. Alors, pour améliorer la gestion des pertes et augmenter les performances du protocole TCP dans l'environnement VANet, il s'avère nécessaire de proposer une nouvelle solution.

Contribution

Dans ce mémoire, nous étudions les possibilités d'améliorer les performances du protocole de transport en réseaux VANet.

Nous montrons l'importance du routage sur les performances des protocoles de transport et en montrant par conséquence l'intérêt d'utiliser les approches de communications entre les couches (Transport - Réseau - Mac).

Les contributions de ce travail sont :

- L'utilisation d'une approche de conception inter-couches (cross-layer) (bottom-up + sublayer).

- Conception d'un nouveau protocole **CVTCP** (**C**ross-layer **V**ANet **T**ransport **C**ontrol **P**rotocol) qui permet de détecter les pertes de paquets, d'identifier avec précision la cause principale de ces pertes (congestion, la mobilité, conditions de canal sans fil, ... etc.), et de résoudre ces différents problèmes en implémentant les mécanismes appropriés.

- l'implémentation d'un nouveau protocole de routage qui permet d'assurer la stabilité de chaque lien, autant que possible, ainsi de garantir la stabilité de la route multi-hop. Cela permet d'optimiser les performance TCP (délai , throuhput) par le meilleur choix du relai (next forwarder).

- L'évaluation de cette solution a été effectuée via un ensemble de simulations afin de tester les performances de notre protocole en modifiant plusieurs paramètres.

Organisation du mémoire

Ce mémoire est organisé en trois chapitres comme suit :

- Le premier chapitre, est une introduction aux réseaux sans fils et en particuliers aux réseaux véhiculaires, leurs domaines d'applications ainsi que les principales caractéristiques de ce type de réseaux. La deuxième partie de ce chapitre est entièrement consacrée aux notions du nouveau concept de cross-layer design.

- Le deuxième chapitre présente des définitions de fonctionnement du TCP ensuite une taxonomie des protocoles de la couche transport dans les réseaux sans fil et Ad

hoc et dans le contexte des réseaux VANets.

– Le troisième chapitre sera consacré aux détails de notre protocole CVTCP proposé dans le cadre de ce mémoire pour les réseaux véhiculaires, et analyse l'efficacité de cette approche par les différentes simulations en utilisant l'outil NS2.

– Enfin, on termine ce mémoire par une conclusion générale et des perspectives.

1 Les réseaux véhiculaires et le nouveau concept cross-layer design

Sommaire

Dans ce chapitre, nous allons présenter le concept des réseaux véhiculaires Ad hoc et leurs caractéristiques ainsi que les différentes applications de ces réseaux.

1.1 Introduction

L'intégration des applications des Technologies de l'Information et de la Communication (TIC) au domaine des transports a permis l'émergence de nouveaux systèmes de Transport Intelligents connus sous le nom ITS (Intelligent Transportation Systems), ces ITS sont étudiés au niveau mondial pour offrir aux passagers des véhicules des services de sécurité routière (alerte accidents, assistance à la conduite,...) et des services de confort (accès à Internet, jeux interactifs,...).

Les réseaux véhiculaires sont une projection des systèmes de transport intelligents. La communication dans les VANets est l'un des domaines de recherche récents qui intéresse de plus en plus la communauté scientifique, les constructeurs automobiles ainsi que les opérateurs des télécommunications. En effet, les systèmes de communication inter-véhicules peuvent être utilisés pour mettre en place plusieurs types d'application appartenant aux systèmes de transport intelligents (ITS) visant à rendre la route plus sûre et de rendre le temps passé sur les routes plus convivial [10].

Dans ce chapitre, nous présentons les concepts des réseaux véhiculaires, ces principales caractéristiques, ainsi que quelques domaines d'applications des VANets. En deuxième partie de ce chapitre, nous allons étudier la notion de *cross-layer design*, et ces principaux types de communication.

1.2 Les réseaux sans fils

Un réseau sans fil est un ensemble d'appareils connectés entre eux via une liaison utilisant des ondes radio et qui peuvent communiquer (envoyer et recevoir des données) sans aucune connexion filaire reliant ces différents composants.

Un réseau mobile sans fil a deux modes de communication, le mode avec infrastructure et le mode sans infrastructure ou mode Ad-hoc, comme il est montré dans la figure1.1

1.2.1 Le mode infrastructure

Cette architecture est la plus utilisée. Elle est particulièrement adaptée pour assurer la connectivité dans des lieux précis tels que dans les aéroports et les hôpitaux. Ce mode repose sur un réseau sur lequel un nœud mobile communique uniquement avec un nœud fixe appelé point d'accès, qui est le responsable des services d'authentification et d'association[11].

1.2.2 Le mode ad hoc

Les difficultés liées à la mise en place des infrastructures réseau ont suscité l'attention des chercheurs à développer des réseaux qu'on peut facilement déployer sans aucune in-

frastructure. La souplesse de déploiement est un avantage majeur dans ce type de réseaux.

Dans le réseau Ad hoc, chaque station opère de manière autonome afin d'assurer sa connectivité avec les autres membres.

Cette architecture est très utilisée dans les scènes qui nécessitent un déploiement rapide et qui prennent en compte la mobilité des stations [12].

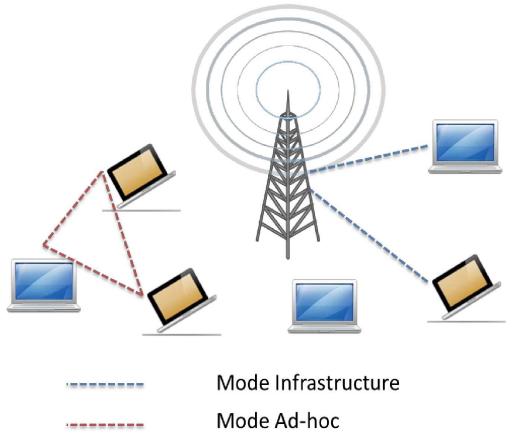

Figure 1.1 – *Communications dans les réseaux sans fils.*

1.3 Les réseaux véhiculaires

Les réseaux VANets (Vehicular Ad Hoc Networks) ne sont qu'une application des réseaux mobiles où les nœuds représentent les véhicules en déplacement. Ces nœuds (véhicules) sont caractérisés par la forte mobilité, de grandes capacités de stockage, une puissance de calcul importante, et une source d'énergie illimitée.

1.3.1 Les architectures de communications dans les réseaux VANets

On distingue principalement trois architectures de communication [13] : ad-hoc (V2V), avec infrastructure (V2I) et hybride comme le montre la figure 1.2.

Communications Véhicule à Véhicule (V2V)

Dans la première architecture (V2V), les véhicules participants forment un réseau mobile ad hoc pour établir des communications entre eux seulement [14].

Communications Véhicule à Infrastructure (V2I)

La deuxième architecture (avec infrastructure) intègre les technologies cellulaires ou autres composants comme les RSUs installés le long des routes pour assurer une communication V2I (Véhicule à Infrastructure) ou I2V (Infrastructure à Véhicule).

Communications hybrides

La troisième architecture est une combinaison de deux architectures précédentes qui permet aux véhicules de communiquer entre eux en mode Ad hoc ou avec un point d'accès (RSU) [12].

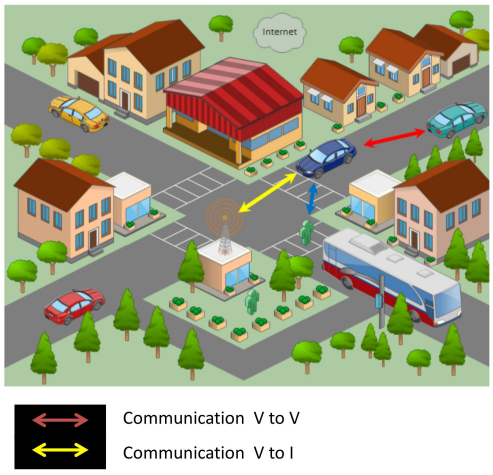

Figure 1.2 – *Architectures de communication.*

1.3.2 Les caractéristiques des réseaux VANet

La communication dans les réseaux véhiculaires est caractérisée par [15] :

- La grande vitesse des véhicules.
- La capacité de stockage.
- La nécessité d'une sécurisation de l'information.
- Les facteurs d'environnement : obstacles, tunnels, congestion etc ...
- Les modèles de mobilité déterminés qui dépendent des conditions de circulation.
- Les ruptures de communication (les réseaux isolés des voitures en raison de la fragmentation du réseau).

1.3.3 Applications

Plusieurs applications peuvent être déployées dans les réseaux VANets. Un consortium d'industriels (General Motors, Daimler Chrysler, Toyota, Nissan, Volkswagen, Ford, BMW) a établi un rapport [16] qui fait maintenant le référence, et qui compte 75 applications. Ces applications peuvent être classifiées en trois classes : La gestion du trafic, le confort, et la sécurité du trafic routier [17].

1.3.3.1 Applications de gestion du trafic routier

Elles consistent à fournir aux conducteurs des informations permettant d'adapter leur parcours à la situation du trafic routier. Les applications de gestion de trafic routier sont axées sur l'amélioration des conditions de circulation dans le but de prévenir la congestion

et les risques d'accidents. En d'autres termes, ces applications visent à équilibrer la circulation des véhicules sur les routes pour une utilisation efficace de la capacité des routes et des carrefours et à réduire par conséquent les pertes humaines, la durée des voyages et la consommation d'énergie. Parmi ces applications on peut citer, la surveillance du trafic, la coopération entre les véhicules afin de faciliter le passage des véhicules d'urgence et l'ordonnancement des feux de signalisation.

1.3.3.2 Applications de confort

Cette classe comporte toutes les applications qui contribuent au confort du conducteur dont l'objectif est de rendre les voyages plus agréables. Ces applications présentent donc des services fournis au conducteur. Parmi ces applications, la gestion des parkings, les jeux/discussions distribués, les applications pair-à-pair, les panneaux d'annonces locales d'ordre commercial comme les offres de restaurants,... etc.

Un autre type d'application de confort est la communication à vocation de divertissement. Une offre de connexion Internet à bord avec vidéo à la demande en est un parfait exemple. À toutes ces applications s'ajoutent aussi les communications point à point entre deux personnes qui voyagent ensemble.

Ils peuvent ainsi s'échanger des messages ou partager des données (vidéo, musique, itinéraire, jeux en réseau). La vie des usagers pourra aussi être facilitée par le contrôle à distance de véhicule de manière électronique (vérification du permis de conduire, contrôle technique, plaque d'immatriculation) pour les services de (police, douane, gendarmerie).

1.3.3.3 Applications de sécurité du trafic routier

L'une des principales motivations du développement et de l'étude des communications véhiculaires est la diminution du nombre de personnes blessées ou tuées sur les routes. Cette catégorie comporte des services qui s'intéressent à l'amélioration de la sécurité routière.

Il s'agit d'améliorer la sécurité des passagers sur les routes en avisant les véhicules de toute situation dangereuse. En lui proposant une aide à la conduite. Le conducteur pourra ainsi anticiper et agir pour rendre la conduite plus sûre. Le conducteur pourra être informé qu'un véhicule vient de passer un feu rouge ou qu'un piéton est en train de traverser la route.

Dans cette catégorie, on trouve les applications qui utilisent les informations des autres véhicules : L'alerte d'état de la route (verglas, obstacle), l'aide au dépassement (calcul des distances, vérification de l'angle mort), l'alerte de freinage ou de collision en amont du trajet. On remarque donc que les applications de sécurité du trafic routier ont un rôle majeur dans la réduction du nombre d'accidents. On remarque aussi que cette catégorie d'applications a des contraintes temporelles fortes. En effet, si l'alerte arrive trop tard, alors le conducteur ne pourra pas éviter le risque. En conséquence, nous perdons les bénéfices de telles applications.

Le Tableau1.1 liste de manière non exhaustive quelques applications des différentes catégories, on remarque que chaque application fonctionne avec un type de communication particulier et un type de message [17]. Le but est donc de développer des protocoles appropriés et de mettre en place des réseaux sans fil véhiculaires composés d'équipements intelligents permettant le déploiement des applications listées.

Application	Communication	Type de message
Feux de freinage d'urgence électronique	Ad hoc V2V	Événementiel, diffusion limitée dans le temps
Alerte de véhicule lent	Ad hoc V2V	Diffusion périodique permanente
Alerte de collision (intersection)	Ad hoc, infrastructure V2I, V2V	Diffusion périodique permanente
Alerte de zone dangereuse	Ad hoc, infrastructure V2I, V2V	Événementiel, diffusion localisée limitée dans le temps
Alerte de violation de feux tricolores	Ad hoc, infrastructure V2I	Événementiel, diffusion limitée dans le temps
Détection pré-accident	Ad hoc V2V	Diffusion périodique, unicast
Alerte de changement de voie	Ad hoc V2V	Diffusion périodique
Alerte coopérative de collision	Ad hoc V2V	Périodique, diffusion événementielle, unicast
Gestion d'intersection	Ad hoc, infrastructure V2I, V2V	Diffusion périodique, unicast
Alerte de déviation	Infrastructure I2V, autre réseau de diffusion	Diffusion périodique
Contrôle de la vitesse de croisière	Ad hoc V2V	Diffusion unicast
Télé péage	Ad hoc, infrastructure V2I, cellulaire	Diffusion périodique, unicast
Diagnostic distant	Ad hoc, infrastructure V2I, V2V, cellulaire	Unicast, diffusion, événementielle
Téléchargement de média	Infrastructure cellulaire, autre réseau de diffusion	Unicast, diffusion, a la demande
Téléchargement de cartes routières	Ad hoc, infrastructure cellulaire, autre réseau de diffusion, V2I, V2V	Unicast, diffusion, à la demande
Assistance de conduite économique	Ad hoc, infrastructure V2I, V2V, cellulaire	Unicast, diffusion, à la demande

TABLE 1.1 – *Exemples des applications pour les réseaux Vanets [17].*

1.4 Activités de standardisation

De nombreuses activités de standardisation portant sur la communication V2V ont été lancées, dans le monde entier, par des organismes internationaux tels que IEEE *(Institute of Electrical and Electronics Engineers)*, ISO *(International Organization for Standardization)*, ETSI *(European Telecommunications Standards Institute)*.

1.4.1 ISO : TC204/WG16 - CALM

Au niveau mondial, le groupe de travail WG16 dans l'organisme ISO a développé une plateforme logicielle embarquée dans les véhicules dite CALM (Continuous Air-interface, Long and Medium range) (voir figure 1.3), dont l'objectif est d'assurer une interface entre plusieurs technologies de communication 2G, 3G, DSRC, ainsi de gérer le handover entre ces technologies.

Par exemple, CALM choisira automatiquement de passer du wifi, du GSM ou du DSRC en fonction de la disponibilité des réseaux et du message à transmettre vers un autre réseau ou technologie. Les applications ciblées vont de la sécurité routière aux applications commerciales, vue la multitude des technologies considérées.

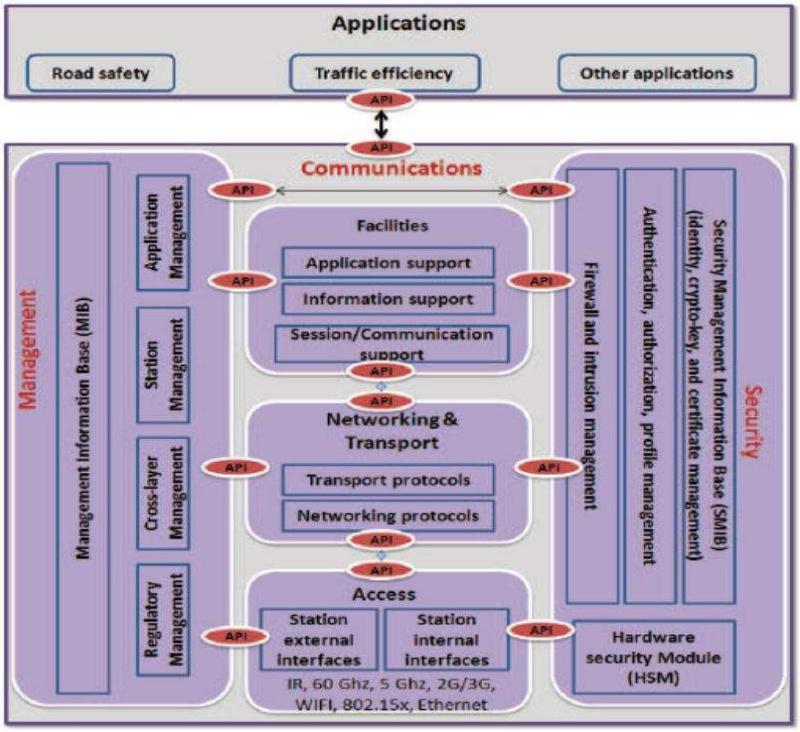

Figure 1.3 – *ISO :CALM [1].*

1.4.2 ETSI : TC ITS

Au niveau Européen, ETSI a créé un comité technique TC ITS, afin de développer des standards et des spécifications pour les ITS. Le comité a mis en place un plan de route afin de produire un ensemble de standards allant de l'architecture de communication à la spécification de protocoles et il est organisé en cinq groupes de travail :

WG1 - User Application Requirements,
WG2 - Architecture and Cross layer issues,
WG3 - Transport and Network,
WG4 - Media and related issues et le WG5 - Security.
Dans le WG3 par exemple, ils s'intéressent à la spécification des protocoles d'adressage et de routage géographiques (voir figure 1.4).

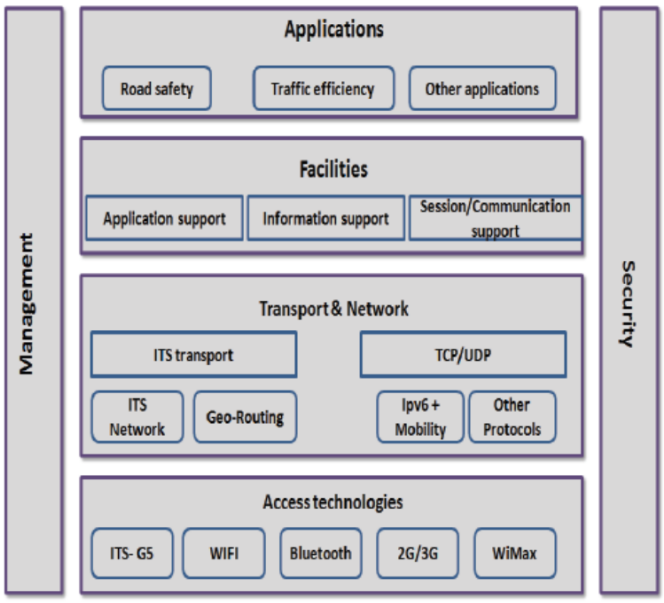

Figure 1.4 – *ETSI :ITS [1].*

La couche **Facilities** est une nouvelle paradigme introduite par l'ETSI et ses principales fonctionnalités sont la fusion et l'entretien des différentes données collectées. La mis à jour en permanence de la carte géographique dynamique LDM (Local Dynamic Map) qui peut être utilisée par les applications, et qui reflète les événements dynamiques qui se produisent sur la route [1].

1.4.3 WAVE (Wireless Access in Vehicular Environments)

L'IEEE a développée une architecture connue sous le nom de WAVE *(Wireless Access in Vehicular Environments)*, pour offrir l'accès sans fil dans les environnements véhiculaires [18].

Nous avons choisi de détailler la pile protocolaire suivant le standard WAVE, La figure1.5 montre l'architecture WAVE qui est une association de l'amendement IEEE 802.11p et de quatre standards 1609.1, 1609.2, 1609.3, et 1609.4 définis par le groupe de travail IEEE 1609 pour décrire les spécifications des couches hautes pour les communications WAVE :

- IEEE 1609.1 WAVE Resource Manager : Pour la gestion des ressources
- IEEE 1609.2 WAVE Security Services for Applications and Management Messages : Pour la sécurisation des messages
- IEEE 1609.3 WAVE Networking Services : Ce standard décrit les fonctions des couches réseau et transport pour les communications dans un système WAVE dont l'adressage et le routage. Il définit un nouveau type de messages dits WSM (WAVE Short Messages) et un nouveau protocole WSMP (WAVE Short Messages Protocol) pour la transmission des WSM. Le WSMP est une alternative à IPv6 qui fournit aux applications un échange de données efficace en leur permettant d'envoyer les WSM directement sur n'importe quel canal de DSRC [2].
- IEEE 1609.4 WAVE Multi-Channel Operation : Pour la coordination et la gestion des canaux de DSRC.

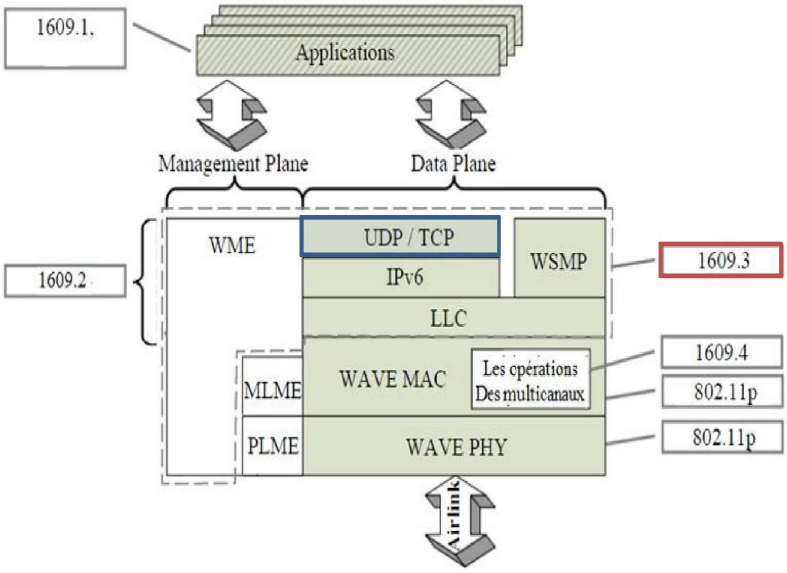

Figure 1.5 – *Architecture protocolaire du système WAVE.*

1.4.3.1 DSRC (Dedicated Short Range Communications)

La FCC *(Federal Communications Commission)* a alloué une bande passante de 75 MHz dans la gamme de fréquences 5,850 - 5,925 GHz pour les communications à courte portée dédiées aux ITS aux USA.

La bande passante est divisée en sept canaux de 10 MHz [19]. Les canaux se répartissent fonctionnellement en un canal de contrôle et six canaux de service. Le canal de contrôle est réservé à la transmission des messages de gestion du réseau tel que les messages d'annonce de services et les messages de très haut priorité comme certains messages critiques liés à la sécurité routière. Les six autres canaux sont dédiés à la transmission des données des différents services annoncés sur le canal de contrôle . Comme le montre la figure 1.6.

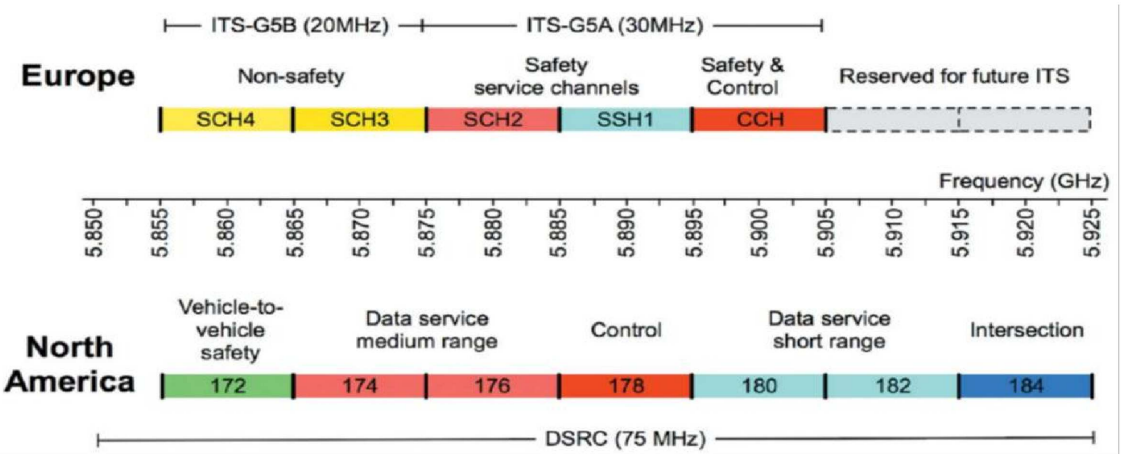

Figure 1.6 – *DSRC [2].*

1.4.3.2 La norme IEEE 802.11p

La norme IEEE 802.11p est un amendement du standard IEEE 802.11 que le groupe de travail IEEE (TGP, task group p) a commencé de développer en 2004 pour l'accès sans fil dans les systèmes de transport intelligents. Il définit les spécifications des couches MAC et PHY dans le cadre des réseaux véhiculaires.[2]

La couche physique du 802.11p utilise les mêmes mécanismes de traitement de signal et les mêmes spécifications que dans le standard 802.11a avec cependant quelques modifications pour l'adapter aux environnements véhiculaires. Pour offrir des communications à grandes portées.

De plus, La couche MAC de la norme 802.11p est équivalente à la technique EDCA*(Enhanced Distributed Channel Access)* de la norme 802.11e. Dans EDCA, les messages sont classifiés en quatre catégories d'accès *(AC, Access Category)*, AC0 la catégorie de messages ayant la plus faible priorité et AC3 la catégorie de ceux ayant la plus grande priorité. À chaque catégorie est associée une file d'attente où ils sont gardés les paquets en attente d'envoi. La priorité est assurée en affectant différents paramètres d'accès à chaque catégorie.

1.4.3.3 la couche application

Le standard IEEE 1609.1 se positionne au niveau de la couche application et définit les formats de messages et le mode de stockage des données utilisées par cette couche [17]. Ce standard définit un gestionnaire de ressources qui autorise des applications de l'équipement de bord de route (RSU) à communiquer avec les On-Board Units (OBU) des véhicules à proximité. Il décrit trois composants de la couche application qui seront inclus dans un OBU :

- *Resource Manager Applications (RMA) :* Entité distante qui utilise le *Resource Manager* RM pour communiquer avec le *Resource Command Processor* (RCP).
- *Resource Manager (RM) :* Le gestionnaire des ressources relaie le message du RMA vers le RCP. Le RM assure les services qui permettent au RMA de contrôler les interfaces présentes dans L'OBU.
- *Resource Command Processor (RCP) :* Il exécute les commandes données par le RMA et fournit une réponse au RMA via le RM.

Lorsqu'une application (présente sur un OBU ou un RSU) veut envoyer une commande à un OBU, le composant RMA envoie un message au RM. Le RM envoie la commande au RCP qui va commander les OBU connectés. Le RCP enverra un message de réponse au RM afin de délivrer le résultat. Le RM est donc le lien entre les applications d'un RSU (ou OBU) et les OBUs des autres véhicules.

1.4.3.4 la sécurité

La sécurité des messages est assurée par le standard IEEE 1609.2 Le but de ce standard est de définir le format des messages sécurisés pour le système DSRC/WAVE. Le standard spécifie les méthodes pour sécuriser les messages de gestion et d'application [17].

1.4.3.5 couches réseau et transport

Dans cette couche il est définit le WAVE Short Message (WSM) et le protocole d'échange associe WAVE Short Message Protocol (WSMP) afin d'assurer les fonctionnalités des couches réseau et transport pour les applications de sécurité routière. D'après la Figure 1.5, la couche réseau utilise le protocole IPv6 pour ses caractéristiques de mobilité, de qualité de service et son espace d'adressage important. En effet, cette dernière caractéristique est primordiale dans un système avec plus de 500 millions de véhicules dans le monde [17].

La nécessité d'un protocole de transport performant : le tableau 1.2 décrit les paradigmes de communication et la nécessité de minimiser le délai et maximiser le throughput pour pas mal d'applications dans les différents types de communications, donc l'utilisation d'un protocole de très haute performance de la couche transport est indispensable, car c'est le seul moyen de garantir telle exigence de qualité de service.

Modes de communication	Maximum débit (throughput)	Minimum délai (delay)
I2V	l'envoi des MP3 video. mises à jour de la carte de navigation. la publicité locale	notification de feux rouges . l'assistant de tourner à gauche
V2I	l'envoi des Emails, la synchronisation de données	l'envoi des notifications d'accident
V2V	l'échange de données (musique,vidéo,...)	avertissement de changement de voie. notification de freinage dur . véhicule de secours approche

TABLE 1.2 – *Modes de communication et applications*

1.5 PROBLÉMATIQUES DE L'ARCHITECTURE EN COUCHES

Le modèle OSI définit une architecture en couches avec une hiérarchie de services. C'est-à-dire que chaque couche réalise un ensemble de fonctions spécifiques en utilisant des services fournis par la couche inférieure et offrant des services à la couche supérieure. Les protocoles dans ces architectures sont conçus de façon indépendante les uns des autres.
Cependant, cette méthode ne conduit pas nécessairement à une solution optimale, notamment pour les réseaux sans fil, et elle a engendrée plusieurs problématiques causées principalement par l'isolation des couches [20]. Ces problématiques peuvent être regroupées en trois grandes classes listées ci-dessous :

- **La redondance** : Elle est causée par la duplication d'un mécanisme sur plusieurs couches. Par exemple, la retransmission dans la couche transport en utilisant TCP et la retransmission dans la couche MAC 802.11.

- **L'annulation** : Les avantages introduits par certains mécanismes de QoS sur les couches supérieures ne sont pas respectés par des couches inférieures ce qui provoque leurs annulations. Par exemple, la priorité introduite au niveau IP grâce aux

classes de service n'est pas forcement assurée au niveau de la couche MAC 802.11.

– **La contradiction** : Dans certains cas extrêmes, l'effet de deux mécanismes présents sur deux couches distinctes est contradictoire. Par exemple, l'utilisation d'UDP au niveau transport pour éviter les retransmissions et l'utilisation de la retransmission au niveau de la couche MAC 802.11.

1.6 Solution

Afin d'apporter une solution à toutes ces problématiques et d'optimiser les performances des systèmes communicants, nous avons assisté ces dernières années à l'émergence d'un nouveau concept sous l'appellation de Cross-layer. Ce dernier autorise la violation de la structure protocolaire en couches dans le but d'améliorer les performances de transmission dans les réseaux sans fil et d'assurer une meilleure QoS.

1.7 Le concept Cross-layer

Comme tous les nouveaux concepts, il est très difficile de trouver ou de proposer une définition exacte pour le Cross-layer. Même au niveau de la terminologie, nous trouvons dans la littérature plusieurs variantes : la conception Cross-layer, l'adaptation Cross-layer, l'optimisation Cross-layer, le retour d'information Cross-layer. Dans [21][22], les auteurs ont proposé une définition générique qui peut englober toutes les techniques et tous les mécanismes Cross-layer qui existent actuellement. Ainsi, le design Cross-layer est défini comme suit : « la conception d'un protocole en violation avec l'architecture en couches de référence est une conception Cross-layer à l'égard de cette architecture ». Le terme violation englobe :

– la définition de nouvelles interfaces entre les couches,
– la redéfinition des limites des couches,
– la conception d'un protocole sur une couche en se basant sur la conception d'un autre protocole sur une autre couche, et
– la configuration commune des paramètres à travers les couches.

1.8 La communication dans les architectures Cross-layer

Le principe de base du concept Cross-layer est de permettre l'échange d'informations entre les couches adjacentes et non adjacentes afin d'améliorer les performances de transmission. Parmi toutes les architectures Cross-layer proposées dans la littérature, on distingue trois architectures de communication [20] :

1.8.1 Architecture Cross-Layer par communication directe

Une première catégorie permet aux couches, même si elles ne sont pas adjacentes, de communiquer directement entre elles (Figure 1.7), afin d'optimiser la Qualité de Service par exemple. Pour cela, il est nécessaire de créer de nouvelles interfaces et d'intégrer de nouvelles routines aux couches qui leurs permettront la réception et le traitement des

données Cross-Layer. Cependant, il faut noter que le nombre de routines à implémenter sera variable selon le nombre de protocoles à satisfaire. Cependant, cette méthode ajoute un certain nombre de contraintes telles que le ralentissement de l'exécution du code [23].

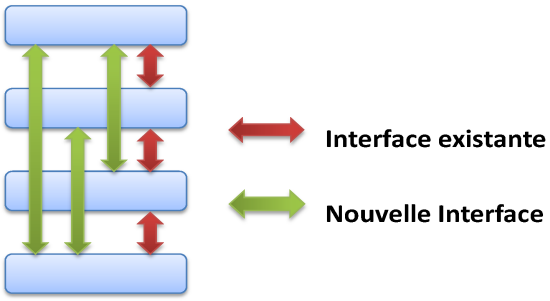

Figure 1.7 – *Architecture cross-layer par communication directe.*

1.8.2 Architecture Cross-Layer avec entité intermédiaire

Une deuxième catégorie permet les interactions inter couches via une entité intermédiaire commune (Figure 1.8). Cette architecture permet de conserver la marche normale de la pile protocolaire, d'où une compatibilité avec l'architecture classique en couches. Cela permet donc de garder tous les avantages inhérents à une architecture modulaire en couches isolées, tels que la robustesse ou la facilité d'évolutivité. De plus, cette méthode permet une évolution continue de l'entité Cross Layer, par l'ajout ou la suppression de protocoles [23].

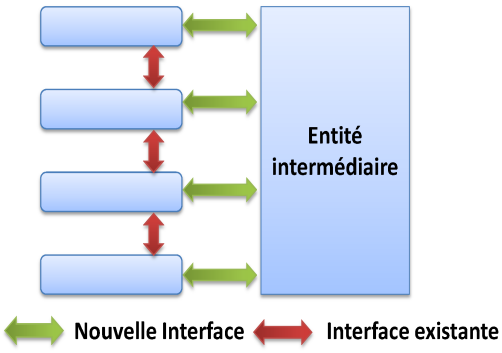

Figure 1.8 – *Architecture cross-layer avec entité intermédiaire.*

1.8.3 Architecture Cross-Layer abstraite du modèle en couche

Une troisième catégorie s'abstrait complètement du modèle en couche (Figure 1.9), elle est donc bien plus flexible, mais elle viole complètement les préceptes du modèle en couches. Il existe des exemples d'architectures Cross-Layer, qui se classent tous dans chacune des trois catégories [23].

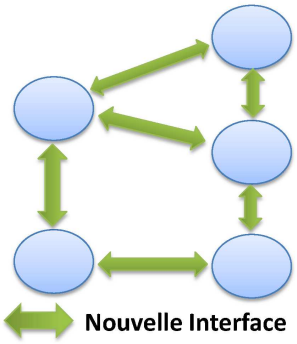

Figure 1.9 – *La troisième catégorie de communication.*

1.9 Les approches du Cross-Layer

Plusieurs techniques Cross-layer ont été proposées dans la littérature pour améliorer les performances de transmissions sans fil. Au début, ces mécanismes étaient limités à l'interaction entre la couche physique et la couche liaison de données. Par la suite, nous avons assisté à l'apparition de plusieurs travaux proposant des interactions avec les couches supérieures, prenant en charge plusieurs paramètres, pour une optimisation globale. Ces travaux peuvent être classés en trois approches identifiées dans [21] [22].

- L'approche ascendante (Bottom-up) où les couches supérieures optimisent leurs mécanismes en fonction des paramètres des couches inférieures.
- L'approche descendante (Top-down) où les couches inférieures considèrent certaines spécificités de niveau supérieur des couches applicatives pour exécuter leurs traitements.
- Et l'approche mixte (Integrated) qui exploite les deux approches précédentes dans une même architecture afin de trouver la meilleure configuration inter-couches pour un fonctionnement optimal du système.

1.10 Conclusion

Les Réseaux véhiculaires (VANET) reçoivent beaucoup d'attention en raison de la grande variété des services qu'ils peuvent fournir. Leurs applications vont de la sécurité et la prévention des accidents jusqu'au multimédia et l'accès à Internet.

Tant que les applications visant à augmenter le confort des passagers en leurs permettant d'accéder à Internet, de télécharger des fichiers ou de jouer en réseau,... etc, sont généralement des applications implémentées dans les couches supérieures, et ont des exigences sur la fiabilité. Ces applications ne peuvent pas être mises en œuvre sans l'utilisation d'un protocole fiable de la couche transport.

Le développement d'un protocole fiable et performant de la couche transport nous a poussé à utiliser et par conséquence à présenter le concept Cross-Layer qui permet de palier les limitations de l'architecture en couche en autorisant un échange d'information entre les couches. Ce nouveau paradigme suscite un grand intérêt pour améliorer les performances des protocoles TCP dans lesquels les conditions du canal radio varient considérablement comparées à celles d'un réseau de type filaire. En effet, le partage de l'état du canal avec les couches supérieures permettra à ces dernières de répondre efficacement à ces changements.

Le chapitre suivant vise à présenter les protocoles de transmission de bout en bout, leurs utilités et leurs fonctionnements, nous allons voir aussi comment exploiter le concept Cross-Layer pour améliorer les performances du protocole TCP dans les réseaux sans fil .

2 Étude du protocole de transport (Transport Control Protocol)

Sommaire

Ce chapitre à pour objectif de décrire et d'étudier les différents approches proposées dans le cadre de l'amélioration de performances de protocole TCP.

2.1 Introduction

L'organisme chargé de standardiser les protocoles du monde Internet, qui sont actuellement adoptés par la majorité des systèmes de communication, propose quatre protocoles de niveau transport : TCP (Transmission Control Protocol) et UDP (User Datagram Protocol), les deux protocoles pères de l'Internet, ainsi que DCCP (Datagram Congestion Control Protocol) et SCTP (Stream Control Transmission Protocol), les deux protocoles fils de la convergence. Les deux premiers ont été spécifiés pour un monde de données alors que les deux derniers, plus récents, prennent en compte les besoins spécifiques au transfert d'information multimédia [24].

TCP (Transport Control Protocol) est un protocole qui a pour but principalement d'assurer l'arrivée des paquets envoyés par un nœud source à la bonne destination, qui gère la congestion du réseau engendrée suite à une émission trop intensive d'informations sur celui-ci et qui contrôle les flux apparus avec des différents débits de transmission entre les nœuds et mener à surcharger leurs buffers. C'est également lui qui se charge de la numérotation et le séquencement des paquets.

Ce chapitre vise à présenter les protocoles de transmission de bout en bout et leurs fonctionnements dans les réseaux filaires et sans fil. La première partie présente les mécanismes de base de ces protocoles. La seconde dresse un aperçu de quelques uns de ces protocoles. En plus, on présente dans ce dernier quelques avantages et inconvénients de chacune des solutions étudiées.

2.2 Le protocole TCP

TCP est un protocole de bout en bout (en anglais Peer to Peer) de la couche transport. Est un protocole en mode connecté, car lorsqu'un canal est ouvert entre un client et un serveur, ce dernier reste valide jusqu'à sa fermeture (qui doit être demandé par au moins l'un des deux applicatifs). Pour identifier un service sur la machine distante TCP utilise les ports. Le numéro de port affecté au client par son système d'exploitation est donc réservé durant toute la connexion TCP. Lorsque deux applicatifs utilisent TCP pour échanger des données, l'émetteur est sûr que le récepteur reçoit exactement les mêmes données envoyées.

TCP assure également la remise dans l'ordre des paquets échangés à l'aide des messages d'acquittement envoyés (ACK). Pour optimiser le transfert, TCP utilise une fenêtre glissante sur le bloc de données qu'il doit envoyer.

2.2.1 Définitions

L'accusé de réception

Est une indication émise de la destination vers la source pour lui signaler que le paquet envoyé est bien reçu. Après un temps précis si l'émetteur ne reçoit pas un accusé de réception pour un des paquets envoyés, il devra le retransmettre (voir figure 2.1).

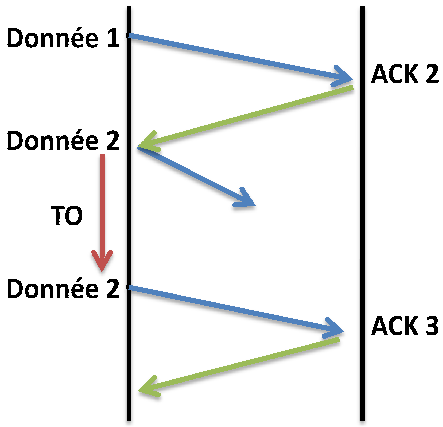

Figure 2.1 – *L'accusé de réception (ACK)* .

Les types d'accusés de réception

Les destinataires peuvent envoyer deux types d'accusé de réception cumulatifs. Ils envoient les acks positifs (ACKs) pour les segments qui sont reçus correctement et dans l'ordre et ils envoient des DUPACKs pour les segments qui sont reçus correctement mais pas dans le bon ordre. Un DUPACK accuse la réception du même numéro de segment que celui accusé par le dernier ACK reçu (voir fig 2.2).

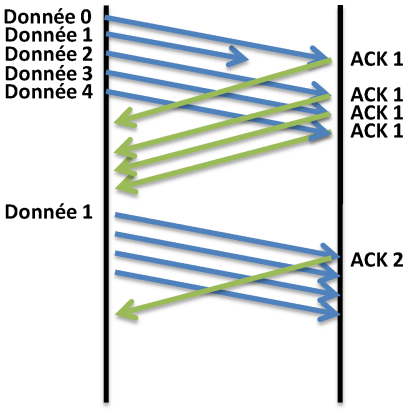

Figure 2.2 – *Les type d'ACK* .

La fenêtre d'émission

La fenêtre d'émission d'un émetteur correspond au nombre de paquets que l'on peut envoyer sur un canal de transmission avant de recevoir obligatoirement un accusé de réception (voir figure 2.3).

La fenêtre de réception

La fenêtre de réception, précise le nombre maximum d'informations que le récepteur est capable de recevoir.

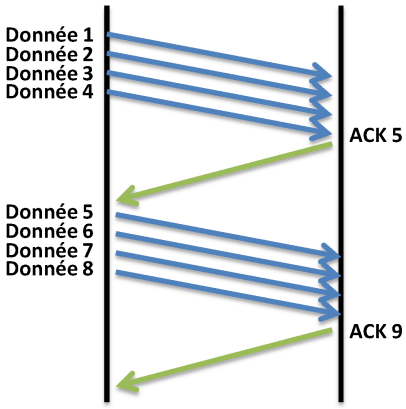

Figure 2.3 – *La fenêtre d'émission .*

Le round trip time

Il s'agit du temps entre l'envoi d'un paquet et la réception de son accusé. Les segments émis par TCP sont de taille fixée par le minimum entre le SMSS (Sender Maximum Segment Size) et le RMSS (Receiver Maximum Segment Size). Cette valeur peut être basée sur l'unité de transmission maximum du réseau (MTU) ou une de ses fonctions.

2.3 Caractéristiques de TCP

TCP apporte des services beaucoup plus élaborés. Ses principaux points qu'il le caractérisent sont :

1. TCP possède un mécanisme pour assurer le bon acheminement des données. Cette possibilité est absolument indispensable dès lors que les applications doivent transmettre de grandes masses de données et de façon fiable. Il faut préciser que les paquets de données sont acquittés de bout en bout. D'une manière générale le réseau assure l'acheminement, et les extrémités le contrôle.
2. Le protocole TCP permet l'établissement d'un circuit virtuel entre les deux points qui échangent de données. On dit aussi que TCP fonctionne en mode connecté (par opposition à UDP qui est en mode non connecté ou encore mode datagramme).
3. TCP a la capacité de mémoriser des données :
   – Aux deux extrémités du circuit virtuel, les applications s'envoient des volumes de données absolument quelconques, allant de 0 octet à des centaines (ou plus) de Mo (voir figure 2.4).
   – À la réception, le protocole délivre les octets exactement comme ils ont été envoyés.
   – Le protocole est libre de fragmenter le flux de données en paquets de tailles adaptées aux réseaux traversés. Il lui rechute cependant d'effectuer le réassemblage et donc de stocker temporairement les fragments avant de les présenter dans le bon ordre à l'application.
4. TCP est indépendant vis à vis des données transportées, c'est un flux d'octets non structuré sur lequel il n'agit pas.

TCP simule une connexion en "full duplex". Pour chacune des deux applications en connexion par un circuit virtuel, l'opération qui consiste à lire des données peut s'effectuer indépendamment de celle qui consiste à en écrire [25].

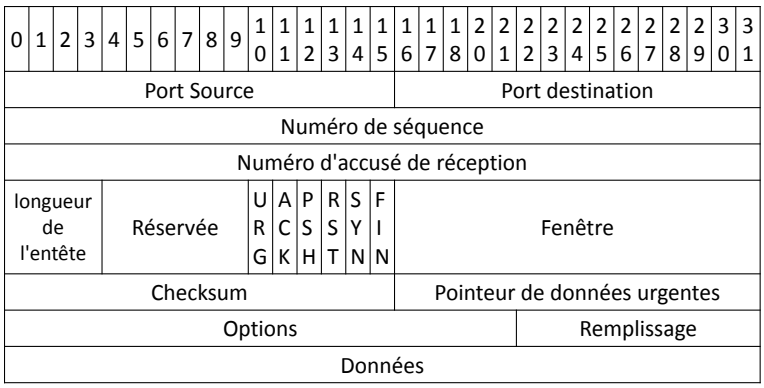

Figure 2.4 – *Format d'un message TCP.*

2.4 L'établissement de connexion dans le protocole TCP

TCP assure la numérotation des paquets, et le destinataire accuse chaque paquet. Il est donc nécessaire pour les deux parties d'établir une négociation du dialogue. C'est pour cela qu'une communication TCP débute toujours par une synchronisation des deux machines.

2.4.1 Ouverture et Fermeture d'une connexion TCP

2.4.1.1 Ouverture d'une connexion TCP

L'établissement d'une connexion TCP s'effectue en trois temps, comme le schéma 2.5 l'explicite [26] :

- L'émetteur envoie un segment comportant le drapeau SYN, avec sa séquence initiale (ISN = x).
- Le récepteur répond avec sa propre séquence (ISN = y), mais il doit également acquitter le paquet précédent, ce qu'il fait avec un accusé de réception ACK (Seq = x + 1).
- L'émetteur doit acquitter le deuxième segment avec ACK (Seq = y + 1).

Une fois achevée cette phase nommée *THREE WAY HANDSHAKE*, les deux applications sont en mesure d'échanger ou de communiquer les octets qui justifient l'établissement de la connexion [27].

2.4.1.2 Fermeture de connexion TCP

Un échange de trois segments (3phases) est nécessaire pour l'établissement de la connexion [26] ; il en faut quatre pour qu'elle s'achève de manière canonique voir fig 2.6.

- L'application qui envoie un paquet avec le drapeau FIN indique à la couche TCP de la machine distante qu'elle n'enverra plus de donnée.
- La machine distante doit acquitter ce segment, comme il est indiqué sur la figure, en incrémentant d'une unité le *SequenceNumber*.

La connexion est effectivement terminée quand les deux applications ont effectuées cette tâche. Il y a donc échange de 4 paquets pour terminer la connexion.

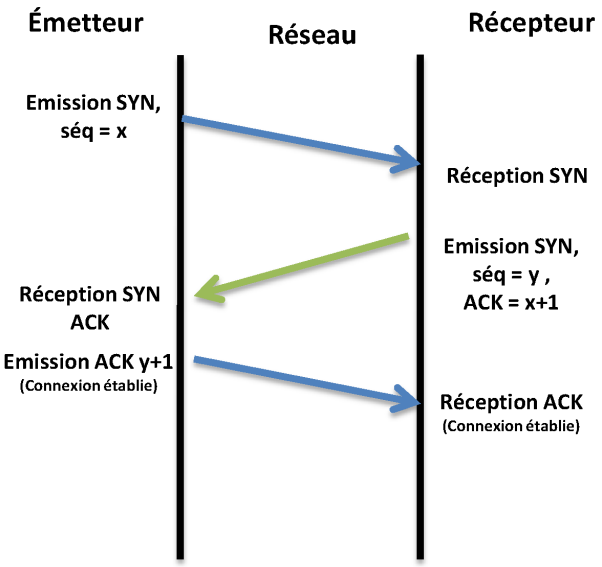

Figure 2.5 – *Ouverture de la connexion TCP .*

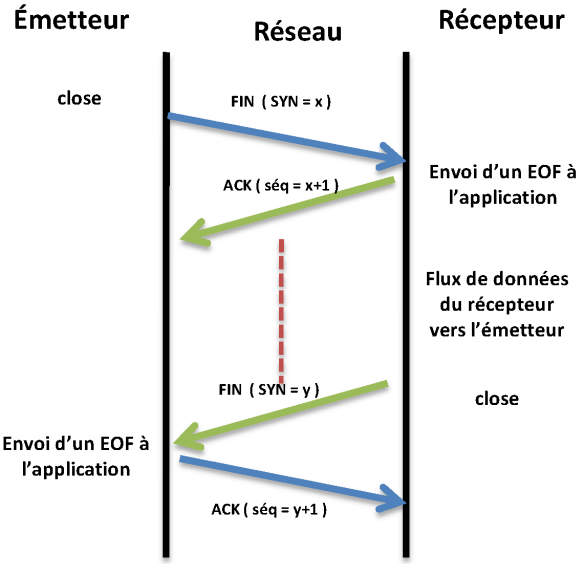

Figure 2.6 – *La fermeture de la connexion TCP .*

2.5 Mécanismes du protocole TCP

2.5.1 Contrôle de flux

Le contrôle de flux est le mécanisme mis en œuvre pour éviter des engorgements du récepteur ; les paquets sont transmis tant que la fenêtre d'émission n'est pas pleine et les acquittements reçus permettent, en vidant la fenêtre, de transmettre de nouveaux paquets.

2.5.2 La fenêtre glissante

TCP régule ses envois à travers un système de fenêtre glissante (*slidingwindow*) voir la fig 2.7 qui définit la quantité de paquets pouvant être envoyés sans être acquittés, en termes de numéro de séquence. Cette fenêtre, dite fenêtre de transmission, que l'on abrègera en

TWND, est calculée à partir de deux autres fenêtres.

- La fenêtre de réception, notée *Rcvwnd*. Elle permet au récepteur d'annoncer le nombre de segments qu'il est capable actuellement de recevoir. L'envoi de cette valeur dans chaque paquet constitue le contrôle de flux.
- La fenêtre de congestion, notée *Cwnd*. Celle-ci est maintenue par le contrôle de la congestion qui s'effectue au niveau de l'émetteur.

La fenêtre de transmission est alors calculée par la formule : Min (Cwnd, Rcvwnd)

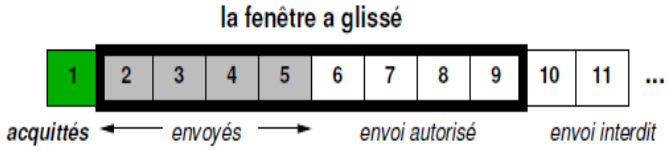

Figure 2.7 – *La fenêtre glissante .*

2.5.3 Contrôle de Congestion

Si le réseau devient congestionné, aucun personne ne peut utiliser les ressources de réseau et le fait que le réseau est congestionné, les paquets transmis ensuite seraient perdus en raison du manque des ressources telles que les espaces de mémoire tampon dans les routeurs. Alors, les terminaux du réseau doivent réagir en cas de congestion par la mise en œuvre des mécanismes de contrôle de congestion afin d'éviter la perte de paquet. Sinon, ils peuvent causer les effets négatifs suivants :

- Augmentation du délai et de la gigue de la transmission de paquets.
- Le délai élevé causé par la retransmission des paquets perdus.
- Réduction du débit du réseau et la mal utilisation des ressources du réseau.

L'objectif principal du contrôle de congestion de TCP est de limiter la fréquence d'envoi afin d'éviter de surcharger le réseau lorsqu'il découvre une congestion sur le chemin à destination.

L'algorithme de contrôle de congestion employé par TCP est basé sur le principe de fenêtres qui sont des espaces de stockage des données limitée par un nombre maximum de paquets [28].

2.5.4 Les phases de démarrage lent et l'évitement de congestion

Les principes de base des algorithmes de congestion *slow start* et *congestion avoidance* sont introduits par Jacobson dans [29].

L'algorithme slow start augmente de façon exponentielle la taille de la fenêtre pour, à l'initialisation, remplir rapidement le réseau de paquets en transit, alors que l'accroissement devient ensuite linéaire, en phase de congestion avoidance, pour s'adapter à la

cadence de sortie de réseau des paquets(voir figure 2.8). Lorsqu'une congestion est détectée, la taille de la fenêtre est diminuée.

La valeur de la fenêtre, sa fonction de croissance avec les seuils pour passer d'un accroissement exponentiel à un accroissement linéaire, font l'objet de plusieurs variantes de TCP. Les différences entre les versions de TCP viennent également des algorithmes appliqués par TCP pour assurer la fiabilité du transfert [24]. Deux modes de recouvrement sont spécifiés dans les standards :

- **Le fast retransmit** : lorsque l'émetteur progresse dans sa transmission de données, mais qu'il n'y a pas de progression dans les acquittements qu'il reçoit, il suppose qu'il y a eu un problème et déclenche la retransmission d'un segment, l'idée étant de ne pas attendre que le temporisateur de retransmission expire pour retransmettre. Le segment qui semble manquer est réémis dès la réception de trois ACKs dupliqués (DUPACKs).

- **Le fast recovery** : consiste à essayer de continuer à transmettre des segments grâce à une diminution adéquate de la fenêtre (afin de ne pas interrompre l'horloge des ACKs qui la font glisser), puis à passer en congestion avoidance [24].

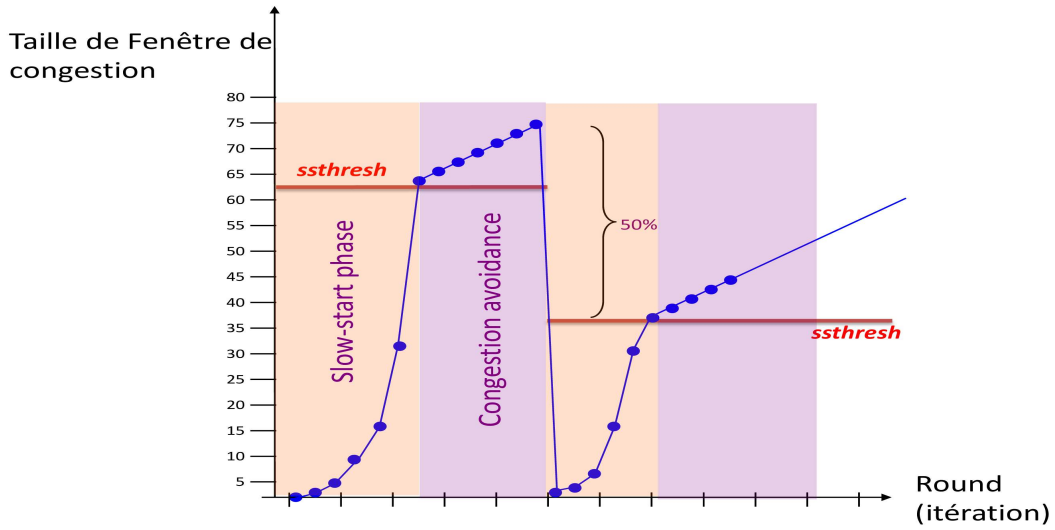

Figure 2.8 – *Démarrage lent et évitement de congestion .*

2.6 TCP dans les réseaux sans fil.

TCP est un protocole de transport qui garantit une transmission ordonnée et fiable des paquets de données sur les réseaux filaires, mais il fonctionne mal en réseaux sans fil, parce qu'il traite toute perte de paquets sur le réseau comme étant le résultat d'une congestion du réseau et il ralentit sa vitesse de transmission. Cela conduit à une dégradation générale des performances. Le développement des réseaux sans fil, rend nécessaire de trouver des moyens d'améliorer l'efficacité et l'utilisation des ressources de TCP, ainsi que de réduire les temps de latence. Afin de trouver des solutions efficaces à cet effet, les pertes de paquets à travers des liens sans fil devraient être distinguées des pertes de paquets à cause de congestion.

Défis du TCP dans les réseaux sans fil

Contrairement aux réseaux câblés, certaines caractéristiques uniques des réseaux réseaux mobiles ad hoc dégradent sérieusement les performances TCP.
Toutes ces caractéristiques posent des grands défis sur TCP pour assurer des communications de bout-en-bout fiables dans ce type de réseau :

- **Multi-sauts :** Dans les réseaux câblés, les routeurs sont des éléments de réseau distincts qui ont la seule fonctionnalité de routage des paquets. Alors que les réseaux adhoc sont des réseaux sans infrastructure, donc les nœuds peuvent jouer le rôle des routeurs c'est-à-dire ils sont responsables de l'acheminement des paquets.
  Et comme inconvénient de cette situation, on trouve que le temps aller-retour (RTO) et la probabilité de perte des paquets deviennent grande, ce qui pose une autre contrainte de l'application du TCP dans les réseaux mobiles en général.

- **La mobilité :** Contrairement à un réseau câblé, dans les réseaux ad hoc sans fil, les nœuds sont libres a se déplacer. Ce qui conduit à des changements fréquents de topologie. Alors deux types d'événements sont possibles :

  - Perte de chemin d'accès ;
  - Partition du réseau.

  La perte du chemin mène l'émetteur à trouver un autre chemin, et au cours de cette phase, il n'y aura aucune transmission, ce qui provoque la dégradation de débit. Au cours de cette phase, la recherche d'un chemin peut prendre beaucoup plus de temps que RTO (timer). Donc, le RTO est augmenté de façon exponentielle et TCP va entrer dans la phase de démarrage lent.
  En cas de partition de réseau, l'émetteur et le récepteur sont considérés appartenant aux réseaux différents et tous les paquets seront perdus.

- **L'occupation de canal :** Une autre raison de dégradation de performance TCP est l'occupation du canal en raison de l'augmentation de nombre des nœuds et la limitation de la bande passante. Dans IEEE802.11, lorsque le nombre d'essaie d'accès au média partagé dépasse la limite prédéfinie, cela peut causer la perte des paquets et le protocole de contrôle d'accès au médium MAC (Medium Access Control) avertit (à tort) la couche supérieure que le chemin d'accès est inaccessible pour que TCP arrête la transmission.

- **Paquet hors ordre :** Lorsqu'un récepteur reçoit les paquets hors ordre, le récepteur transmet ACK doublé. Après avoir reçu trois ACK doublés l'émetteur retransmet les paquets et le contrôle de congestion est activé. Mais le problème, c'est que le contrôle de congestion est activé à tort la plupart du temps, parce que la présence de paquets hors ordre aura lieu pour des raisons différentes telles que le protocole de routage multi-chemins et la rupture des liens et pas seulement à cause de congestion.

- **La déconnexion :** Dans les réseaux sans fil en général les déconnexions peuvent arriver à cause de plusieurs raisons :

  - Quand le nœud mobile bouge d'une cellule à une autre. Cette procédure de réassociation, nommée *Handover*, qui dépend fortement de la technologie de transmission

utilisée, peut provoquer de courtes interruptions de la connectivité. Le protocole du transport assimile ces périodes de déconnexion à des phases de congestion et déclenche inutilement la procédure de contrôle de congestion qui diminue le débit du flux.
- Quand un hôte mobile bouge de la portée des émetteurs-récepteurs.
- Quand les signaux radio sont bloqués par les bâtiments et d'autres objets semblables.
- Quand une cellule contient un grand nombre d'utilisateurs et la bande passante n'est pas suffisante pour satisfaire leurs besoins.

Toutes ces caractéristiques et d'autres problèmes comme l'interférence et les problèmes des terminaux cachés et exposés ne permettent pas l'application du TCP classique dans ce type de réseaux. En raison de l'usage commun du TCP dans les applications Internet qui exigent un transfert fiable de données, il est important de modifié le TCP classique afin de satisfaire les attentes de rendement TCP dans l'environnement sans fil.

.

2.7 Taxonomie

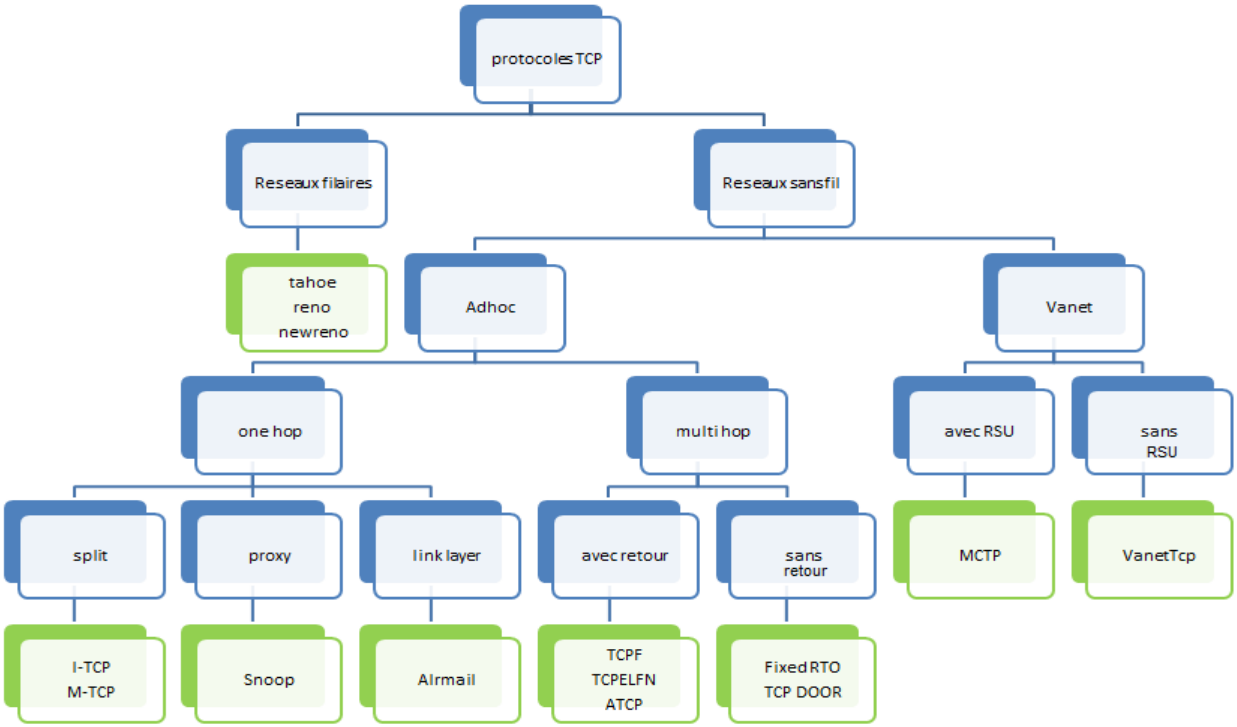

Figure 2.9 – *Les Protocoles de Transport* .

2.7.1 Pour les réseaux avec infrastructure

Les solutions proposées pour améliorer les performances du protocole de transport, (dont une vue globale est présentée en [30]) peuvent être regroupées en trois catégories : découpage de la connexion(split connection), utilisation de mandataires de service transport(proxy), contrôle de niveau liaison (link layer) voir la figure 2.9.

2.7.1.1 Découpage de la connexion

1. **TCP indirect (I-TCP)** L'idée du protocole I-TCP est de diviser explicitement la connexion TCP entre l'hôte fixe (FH) et le hôte mobile (MH) en deux connexions une filaire entre l'hôte fixe et le point d'accès et l'autre sans fil entre le point d'accès et l'hôte mobile [4].

   – Quand un utilisateur mobile souhaite communiquer avec un utilisateur fixe. Le protocole I-TCP résidant sur le mobile émet une demande d'établissement de connexion TCP, pour l'utilisateur fixe, à le point d'accès. Concernant l'acquittement, un paquet transmis du poste fixe vers le mobile est d'abord acquitté sur le premier tronçon de connexion par la station de base, puis est transmis sur le second tronçon voir la figure 2.10. Avec cette solution, les acquittements n'ont plus de signification de bout en bout, chaque connexion gérant ses propres acquittements.
   – Dans le cas d'un déplacement à une nouvelle cellule (Handover).
   – le MH établit une connexion avec le FH via l'ancien point d'accès PA-1, et ensuite il se déplace vers une autre cellule sous le PA-2.
   – Si le MH demande une connexion I-TCP avec le FH dans la zone de PA-1, celui la établit un socket avec adresse et numéro de port du MH. « Mhaddr, mhport, fhaddr, fhport », ensuite il crée un autre socket avec ces propres adresses et numéro de port. « pa1addr, pa1port, mhaddr, mhport ».
   – Quand le MH change de cellule, l'état de deux sockets de la connexion I-TCP au niveau du PA-1 est remis au nouveau PA-2, ce dernier crée deux sockets correspondant à la connexion I-TCP avec les mêmes paramètres de l'extrémité que celle des sockets au niveau PA-1.
   – Puisque les extrémités de la connexion pour le mobile et le fixe de I-TCP ne changent pas après le mouvement, il n'est pas nécessaire d'établir une connexion avec le nouveau PA.

   Avantages :
   – Pas besoin de modifier les TCP existants dans le réseau fixe.
   – Les erreurs des liens sans fil sont corrigées par le point d'accès et ainsi ne se propage pas dans le réseau jusqu'à la source.
   – Le nouveau protocole a un impact sur une partie limitée de l'Internet (sans fil).
   – Des optimisations sont possibles sur les liens sans fil.
   – La variance du délai entre le point d'accès et le nœud mobile peut être petite permettant ainsi d'optimiser TCP.

   Inconvénients :
   – Perte de la sémantique de bout en bout de TCP. Que se passe-t-il si le Proxy ou le nœud mobile tombe en panne ?
   – Le surcoût de traitement du handover peut s'avérer important.
   – Le surcoût de traitement par paquet au niveau du point d'accès peut s'avérer important [27].

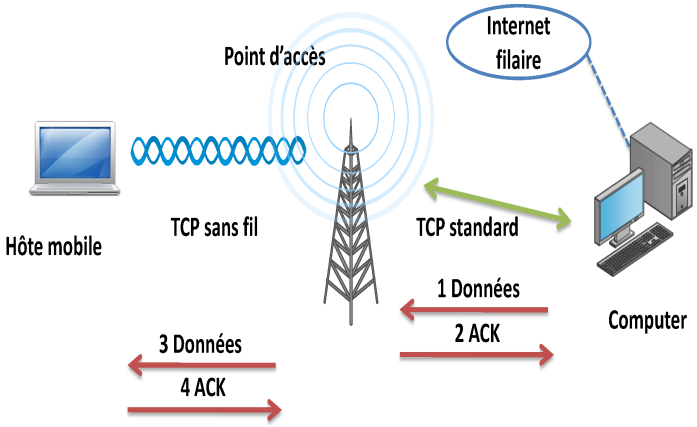

Figure 2.10 – *Le protocole I-TCP.*

2. **Mobile TCP (M-TCP)** M-TCP c'est une autre approche similaire à I-TCP qui coupe la connexion TCP en deux mais avec une différence avec I-TCP, est que celui-ci préserve la sémantique de TCP de bout en bout. M-TCP fonctionne avec une architecture à trois niveaux [31].

   – Au niveau le plus bas on trouve les nœuds mobiles (MH) qui communiquent avec les MSS (Mobile Support Station) de la même cellule.
   – Un groupe de MSS est contrôlé par un hôte superviseur (SH), qui joue le rôle d'un routeur, et il est connecté au réseau fixe voir la figure 2.11.
   – La plupart optent pour cette architecture pour deux raisons :

     – la première c'est que les fonctionnalités au niveau de MSS peuvent être transférées au SH qui peut réduire le coût de réseau car un SH a en charge plusieurs MSS.
     – la deuxième est que le nombre de Handovers est très réduit étant donné que le déplacement de MH d'une cellule à une autre n'a pas besoin d'effectuer des Handovers, tant que les deux cellules sont contrôlées par le même SH.

   Avantages :
   – Maintient la sémantique TCP de bout en bout
   – L'efficacité de la connexion TCP n'est pas dégradée en raison de la déconnexion car l'émetteur n'empêche pas à entrer à un backoff exponentiel et Slow Start.

   Inconvénients :
   – Les pertes de la liaison sans fil se propagent dans le réseau fixe.
   – L'adaptation de TCP dans le lien sans fil [27].

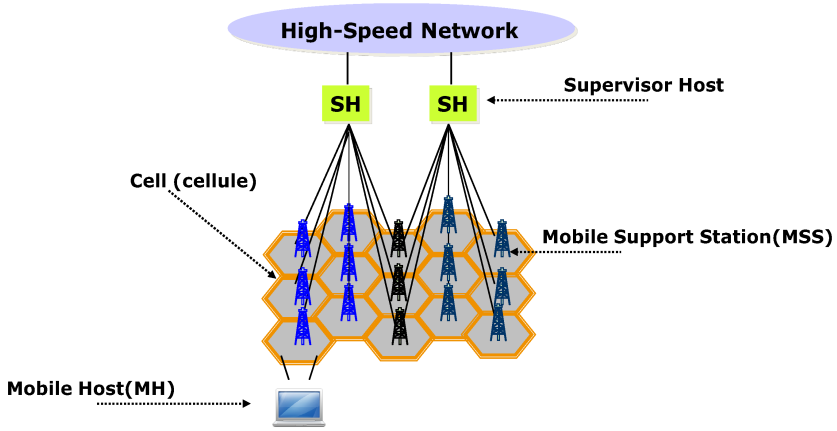

Figure 2.11 – *Le protocole MTCP.*

2.7.1.2 TCP avec mandataire de service (proxy)

1. **TCP SNOOP** Une autre approche qui préserve la sémantique de bout en bout et qui modifie les mécanismes de la couche transport au niveau de la station de base par l'ajout d'un module « SNOOP » « espion » qui contrôle les paquets passant par cette connexion dans les deux sens [32].

   Le module SNOOP maintient un cache des paquets TCP envoyés par le FH et non encore acquittés par le (MH), à l'arrivée d'un nouveau paquet de FH voir la figure 2.12, le module SNOOP l'ajoute à son cache et le route vers le MH approprié dans la cellule appropriée. Il garde aussi une image de tous les acquittements transmis par le MH. Si il y a une détection de perte de paquet (arrivée de 3 ACKs successifs ou par l'expiration de RTO local), le module SNOOP retransmet le paquet perdu au MH (dans le cas où le paquet a été caché). Ainsi le SNOOP cache les paquets perdus au FH par la non propagation des acquittements dupliqués DUPACKs, ce qui permet d'éviter l'invocation inutile de mécanisme de contrôle de congestion.

   Dans le cas où les paquets sont envoyés par le MH vers le FH, le MH ne détecte pas la cause de perte (erreur de lien, congestion réseau), alors il utilise l'option TCP SACKs. Quand le module SNOOP remarque un espace à l'intérieur de numéro de séquence de paquet transmis par le MH, alors le SNOOP envoie un SACK-enable pour le MH qui retransmet le paquet perdu. Le module SNOOP utilise deux procédures pour traiter les cas précédents :

   Snoop data () : traite les paquets reçus de FH.
   Snoop ack () : contrôle et traite les acquittements envoyés par le MH.

   Snoop data() traite les paquets de l'émetteur de la manière suivante :

   – Lorsqu'un paquet (ayant un numéro de séquence) arrive, il est sauvegardé et transféré au récepteur.
   – Lorsqu'un paquet (ayant un numéro hors-séquence (en désordre)) qui est déjà été sauvegardé, représentant une retransmission d'un paquet perdu par congestion) arrive, il est tout simplement transféré au récepteur.

– Lorsqu'un paquet (ayant un numéro hors-séquence que n'est pas encore sauvegardé) arrive, il est marqué comme congestion dans le tampon de l'AP et puis transféré au récepteur.

Snoop ACK() en revanche surveille et traite les accusés de réception (ACKs) et effectue différentes opérations selon le type et le nombre d'accusés de réception qu'il reçoit . Ces accusés de réception peuvent tomber dans l'une des trois catégories :

– un nouvel ACK.
– un ACK parasite.
– Un DupACK.

Quand un ACK est reçu : premièrement, si c'est une nouvel ACK, son paquet est libéré et puis l'ACK est transféré à l'émetteur. deuxièment , si c'est le premier DupACK, le paquet et directement retransmis localement. Dans tous les autres cas, l'ACK est ignoré.

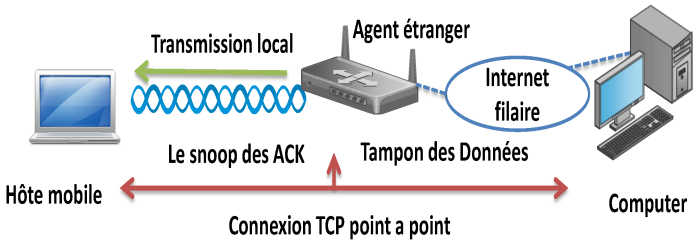

Figure 2.12 – *Le protocole Snoop TCP.*

Avantages :
– Préserve la sémantique de bout en bout.
– Performant dans un environnement à BER élevé.
– Utilisation du protocole de la couche liaison pour le recouvrement des pertes.

Inconvénients :
– Si le terminal portable se déplace vers une station de base ne faisant pas partie du groupe, reconstruire le cache prend alors un temps important. Cette méthode induit néanmoins un coût important sur le réseau fixe [27].

2.7.2 TCP dans les réseaux Ad hoc

2.7.2.1 Avec retour d'information (Feedback)

1. **TCP-F « TCP FEEDBACK »** Une approche appelée TCP-F (TCP avec retour d'information) a été proposée pour que le TCP émetteur (sender) peut distinguer entre l'échec de route (Route Failure) et la congestion du réseau[33].

   – À la détection d'une interruption du route, il envoie explicitement un paquet de notification d'échec de route (Route Failure Notification) (RFN) à l'émetteur et enregistre cet événement.
   – À la réception du RFN, la source (émetteur) devient dans un état de *snooze* (sommeil) dans lequel il stoppe complètement la transmission des paquets, et gèle tous les variables tels que les valeurs de RTO et la taille de la fenêtre de congestion (Cwnd).
   – L'émetteur reste dans l'état *snooze* jusqu'au être notifié par la restauration de route via un paquet de notification de rétablissement de route (Route Reestablishment notification) (RRN) à partir d'un nœud intermédiaire.
   – À la réception du RRN, le TCP émetteur quitte l'état de *snooze* et reprend la transmission à partir des valeurs de RTO et Cwnd précédemment enregistrées comme il est montré dans la figure 2.13.

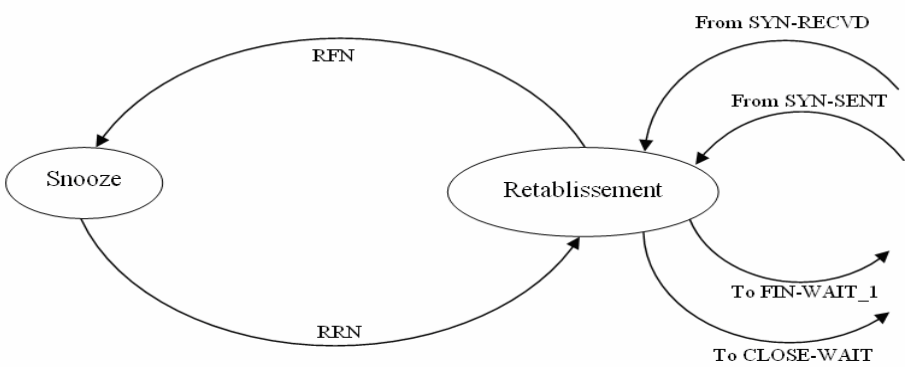

Figure 2.13 – *Le protocole TCP-F.*

2. **TCP-ELFN « Explicit Link failure Notification »** Une autre technique similaire à TCP-F, basée sur le retour d'information « Feedback », notification explicite de l'échec de liaison. Cependant à l'opposé de celle-ci, l'évaluation de cette proposition est basée sur l'interaction réelle entre le TCP et le protocole de routage. Cette interaction vise à informer l'agent TCP sur les échecs de route quand celles-ci se produisent [6]. Cette technique est basée sur le protocole de routage DSR.

   – Pour exécuter le message ELFN , le message d'échec de route de protocole DSR est modifié afin de porter un poids utile de charge similaire au message ICMP.
   – À la réception du ELFN, le TCP émetteur désactive ces mécanismes de contrôle de congestion et entre dans un mode *Stand By*, qui est similaire à l'état *snooze* du TCP-F.
   – À l'opposé du TCP-F qui utilise une notification explicite pour signaler la restauration d'une route, l'émetteur , pendant la période de *Stand-by* ,investigue « explore » le réseau pour vérifier si un chemin est rétabli (restauré). Si un paquet d'acquittement de l'investigation est reçu, le TCP émetteur quitte l'état de Stand-by

et reprend ces Timers de retransmission (RTO) et continue ses opérations normales.

Avantages :
– TCP-EFLN et TCP-F permettrent à l'émetteur d'entrer immédiatement en état de *snooze* et d'éviter la retransmission inutile et le contrôle de congestion.

Inconvénients :
– Aucun de deux protocoles ne considère l'effet de la congestion, le désordre des paquets et les erreurs de bit (BER) qui sont complètement commun dans les réseaux sans fil ad hoc.
– Les deux utilisent les mêmes séries de paramètres incluant la taille de la fenêtre de congestion et le RTO après le rétablissement des routes comme celles avant l'échec de route, qui cause un problème parce que la taille de la fenêtre et la valeur de RTO ne sont pas identiques d'une route à une autre surtout dans le cas où le changement de route est significatif [27].

3. **Ad hoc TCP (ATCP)** Une approche appelée ATCP essaye de s'occuper du problème BER utilise aussi le feedback de la couche réseau[3]. En plus de l'échec de route, le TCP émetteur peut se mettre dans l'un des états suivants :

   – Etat persistant : perte de paquets due à un échec de route
   – Etat de contrôle de congestion : perte de paquets due à une vraie congestion de réseau.
   – Etat de retransmission : perte de paquets due à un taux d'erreur de bit élevé.

   Une couche fine, appelée ATCP est insérée entre la couche TCP et IP du nœud source. ATCP lit l'information de l'état du réseau donnée par le message "ECN" (Explicit Congestion Notification) et le message "ICMP" et par la suite, l'ATCP met l'agent TCP dans l'état approprié.

   – A la réception d'un message "ICMP", l'agent TCP entre dans un état persistant, durant lequel l'agent est gelé et aucun paquet n'est transmis jusqu'à qu'une nouvelle route est retrouvée par l'exploration du réseau, et donc l'émetteur ne peut pas invoquer un contrôle de congestion.
   – La réception d'un message "ECN", invoque un contrôle de congestion sans attendre l'événement de Time Out pour détecter les pertes de paquets dues à une erreur de transmission.
   – ATCP contrôle les acquittements reçus, quand ATCP remarque qu'il a reçu trois (03) acquittements dupliqués, il met le TCP dans l'état persistant et retransmet rapidement le paquet perdu à partir du buffer. Après la réception d'un acquittement non dupliqué, ATCP met TCP dans un état normal.
   – Si une perte de paquet se produite et si le champ ECN n'est pas à jour, ATCP suppose que cette perte est due à une erreur de bit et il retransmet simplement le paquet perdu.

   L'automate de la figure 2.14 schématise l'état de transition de l'ATCP au niveau émetteur :

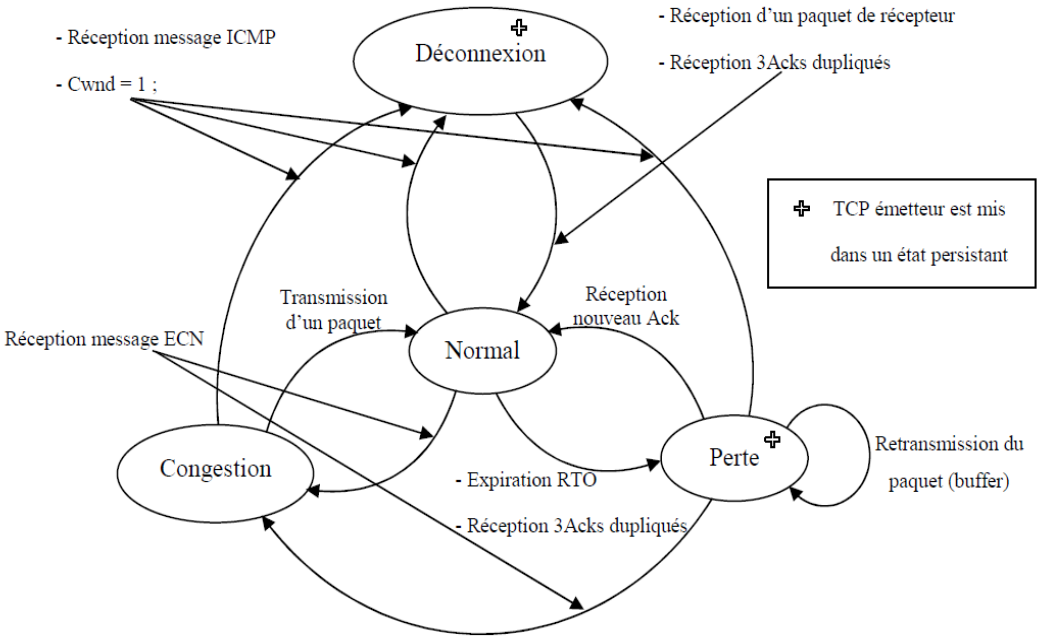

Figure 2.14 – *Le protocole ATCP [3].*

4. **ADhocTCP (ADTCP)** La conception clé de ce protocole proposé par Zhenghua Fu et al [34] est d'utiliser une jointure de multi-métriques pour détecter l'événement réseau et mettre le TCP dans l'état approprié.

   Les métrique utilisées sont classées dans le tableaux 2.1 :

   – IDD : Inter-packet delay difference.
   – STT : Short-term throughput.
   – POR : Packet out-of-order delivery ratio.
   – PLR : Packet loss ratio.

TABLE 2.1 – *Résumé des métriques utilisées*

Métriques	Définition
IDD	$A^i + 1 - A^i - (S^i + 1 - S^i)$, ou A^i c'est le temps d'arrivé du paquet i et S^i c'est le temps d'envoi
STT	$N_p(T)/T$, ou $N_p(T)$ c'est le nombre de paquets reçus durant un intervalle T
POR	$N_{po}(T)/N_p(T)$, ou $N_{po}(T)$ c'est le nombre de paquets hors ordre durant T
PLR	$N_l(T)/N_p(T)$, ou $N_l(T)$ c'est le nombre de paquets perdus durant T

Le tableau 2.2 résume le comportement de protocole et le passage entre les cinq états du réseau :

TABLE 2.2 – *Résumé des états du réseau*

	IDD et STT	POR	PLR
CONGESTION	(High, Low)	*	*
ROUT CHANGE	NOT (High, Low)	High	*
CHANNEL ERRORE	NOT (High, Low)	*	High
DISCONNECTION	(*,$\simeq$ 0)	*	*
NORMAL	default		

2.7.2.2 Sans retour d'information

1. **Fixed RTO** Dans la solution TCP à timer fixe,"Fixed RTO" , aucun retour d'état n'est utilisé [35]. Après une expiration du timer, si l'émetteur ne reçoit pas d'acquittement avant l'expiration du timer une deuxième fois, l'émetteur déduit que le réseau n'est pas congestionné mais qu'il y a un problème de routage ; il retransmet le paquet perdu mais n'augmente pas la valeur du timer (RTO). Le RTO reste fixe jusqu'à la rétablissement de la route [24].

2. **TCP DOOR** (Detection of Out-Of-Order and Response) est proposé par Wang et Zhang [36]. L'objectif est d'améliorer les performances sans utiliser de retour réseaux, mais en étant capable de détecter et de réagir à des changements de routes. Le protocole assimile les changements de routes à des évènements de hors séquences (out of order : OOO). Ces hors séquences peuvent être fréquents dans les réseaux MANETs en raison de la mobilité des nœuds et des chemins. La détection du hors séquence est effectuée, à l'émetteur, sur les acquittements et, au récepteur, sur les données, grâce à une nouvelle numérotation (dans le TCP de base le numéro n'est pas incrémenté lors des retransmissions)[24].

   – Lorsque le récepteur détecte, grâce au nouveau numéro TPSN (TCP Packet Sequence Number), un hors séquence sur les données reçues, il renvoie l'information à l'émetteur en positionnant un flag dans l'acquittement correspondant.
   – De même, l'émetteur vérifie le flux retour grâce à une numérotation spécifique dans l'acquittement.
   – En cas de détection de hors séquence, l'émetteur dispose de deux mécanismes, l'un permet, comme dans TCP-Feedback, de bloquer le contrôle de congestion, car il y a une perte de route momentanée, l'autre d'accélérer momentanément le recouvrement d'erreur. Si le contrôle de congestion a été récemment effectué et qu'un évènement de hors séquence est détecté, l'émetteur déduit qu'il y avait une coupure momentanée de la connexion et non pas une congestion.

2.7.3 TCP dans les réseaux VANets

La conception d'un Transport Control Protocol (TCP) dans VANets est une tâche difficile, car la transmission de données de bout en bout est une transmission sans fils qui caractérisée par les sauts multiples et la forte mobilité.

Afin d'accéder aux services d'Internet, les architectures VANet doivent comporter des proxies, les véhicules vont se connecter à Internet via ces proxies. Un proxy est localisé à une position fixe et peut cacher les caractéristiques des VANet, donc il sépare également la connexion TCP de bout en bout en deux segments, entre le proxy et Internet en utilisant

TCP classique (standard), et entre les véhicules et le proxy, en utilisant des protocoles de transport optimisés afin d'améliorer l'efficacité de communication.

les caractéristiques spécifiques aux réseaux VANets posent des grands challenges pour le TCP afin d'assurer des communications de bout-en-bout fiables.

2.7.3.1 VANet TCP

En se basant sur les connaissances acquises à partir de protocole ATCP (Ad-hoc TCP)[3], un nouveau protocole de couche de transport a été proposé et implémenté par Wantanee Viriyasitavat, et al dans [7] sous le nom de VANet TCP. Les contributions spécifiques de ce travail sont :

- la dissimulation du problème de contrôle de transmission de bout-en bout comme étant un problème de couche réseau (routage). Il est clair que pour certaines applications, la livraison réussite des paquets est plus souhaitable que le fait de pouvoir transmettre un paquet en un temps réduit. Donc, une route plus stable devrait être choisie même si elle est plus longue.

- L'utilisation d'une approche de conception inter-couches (cross-layer) dans lequel l'information de la couche de transport est utilisée par la couche réseau afin de permettre la prise des meilleures décisions par chaque nœud. Cette flexibilité permet à la couche réseau de choisir dynamiquement la route la plus appropriée pour la couche de transport.

- La conception d'un nouveau protocole de Transport qui permet de détecter les pertes de paquets, d'identifier la cause principale de ces pertes correctement (congestion, la mobilité, conditions de canal sans fil, ... etc) et de résoudre ces différents problèmes avec les mécanismes appropriés.

Comme la FIGURE 2.15 le présente, VANet TCP peut distinguer trois types de perte de paquets :

- **Paquet perdu à cause de la congestion**

  À la réception d'un acquittement marquée par un bit d'avis de congestion explicite (ECN), et si le système est dans l'état NORMAL il va basculer vers l'état de CONGESTION, et pour chaque ACK reçu, la couche de transport devrait réduire le débit de transmission (la fenêtre de congestion est réduite par la moitié) avant de continuer la transmission des données.

  Le système revient à l'état NORMAL s'il reçoit un nouveau ACK avec le bit ECN démarqué (c.-à-d. la congestion a été résolue).

- **Paquet perdu à cause de la charge du canal sans fil**

  À la réception des acquittements doublés, et si le système est à l'état NORMAL, il va basculer vers l'état de ERREUR CANAL, où il doit sauvegarder le dernier nombre de séquence des paquets transmis, noté par *Seqnochannel*. Dans cet état, le protocole envoie seulement les segments perdues qui sont indiquées par les ACKs doublées

(transmission de toutes les données autorisées par la fenêtre de congestion sans réduction du débit de transmission ou de la taille de la fenêtre de congestion).

Le système revient à l'état NORMAL s'il reçoit un nouveau ACK avec un nombre de séquence plus grand que *Seqnochannel*, attestant que tous les paquets perdus ont été recouvrés.

– **Paquet perdu à cause de déconnexion.**

À l'expiration du RTO, le système va basculer à l'état de DÉCONNEXION quelque soit l'état courant. Le système doit sauvegarder les paramètres courants (telle que taille de fenêtre de congestion, RTT etc ...), ensuite il arrête la transmission des données et commence à envoyer des paquets périodiques de sonde au réseau jusqu'à qu'une nouvelle route est découverte.

À la découverte d'une nouvelle route (indiquée par la réception d'ACK), l'état doit basculer à NORMAL en restaurant les valeurs des paramètres sauvegardés avant la déconnexion.

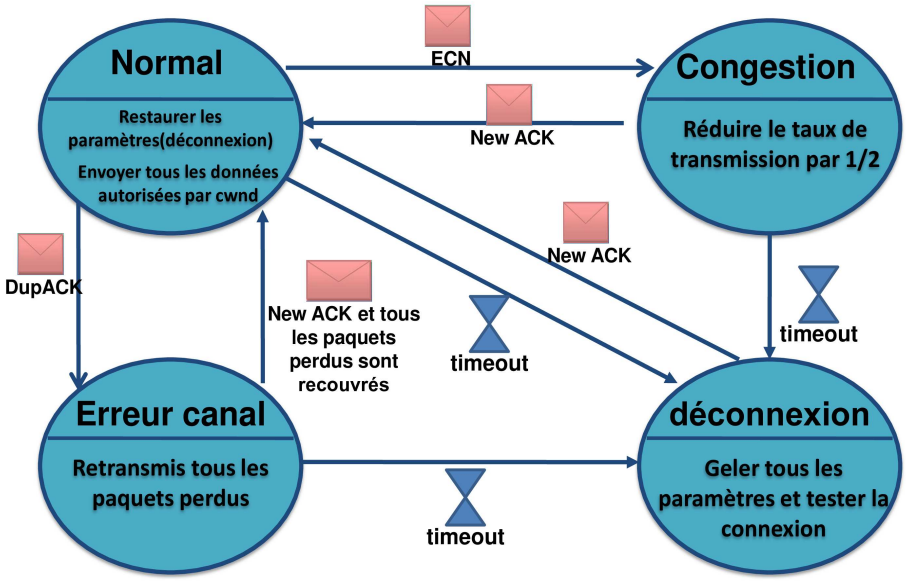

Figure 2.15 – *Le protocole VanetTCP.*

Les Interactions Inter-Couches : Le mécanisme de cross-layer exige dans ce protocole que la couche transport informe la couche réseau sur la qualité de la route souhaitée.

L'information utilisée entre la couche de transport et la couche réseau s'appelle EPDR(t) (Expected Packet Delivery Ratio). À la réception de cette valeur, la couche de réseau arrange son processus de sélection d'acheminement, comme le fait le protocole : Beacon Reception Rate routing protocol (BRR) adopté dans ce travail.

Le protocole de routage Beacon Reception Rate (BRR) :

C'est un exemple d'un protocole de routage basant sur la connaissance de l'état de la couche transport (en anglais transport-aware routing protocol) où il est adopté selon l'information reçue de la couche de transport d'une façon adaptative et dynamique.

Au lieu de choisir la route avant la transmission de données, le véhicule source et les autres véhicules de relais choisissent un de leurs voisins pour transmettre le paquet de données vers la destination. Ces dernières sont choisis suivant leurs localisations et leurs états de connexion avec le véhicule source et les autres relais.

Dans le protocole de BRR, un véhicule garde en plus de l'adresse et l'information de localisation, la valeur (BRR) de chacun de ses voisins. Cette information de BRR est utilisée par un véhicule comme indicateur principal pour estimer la qualité de son lien avec leur voisin et pour déterminer le meilleur relai.

Le calcul du BRR se fait à des intervalles fixes et connus, le véhicule i peut calculer le BRR de son véhicule voisin J au moment T, noté par $BRR_i(j,t)$ de la façon suivante :

$$BRR_i(j,t) = \frac{NMR_j[t-w;t]}{NME_j[t-w;t]}$$

Où le w est la durée de la fenêtre de BRR durant laquelle les statistiques sont rassemblées, et NMR_j c'est le Nombre des messages reçu de j pendant $[t-w;t]$ et NME_j Nombre des messages émis par j pendant $[t-w;t]$.

Le choix du prochain véhicule relai :

En supposant que le véhicule i est le prochain véhicule sélectionné comme l'indique l'entête d'un paquet, alors, si le véhicule i recevra un paquet destiné au véhicule K dans le temps T, va consulter sa table des voisins afin de choisir le prochain véhicule qui doit faire suivre le paquet. S'il y a un véhicule j dit qu'elle est le plus proche à i de la destination avec une valeur de $BRR_i(j,t)$ plus grande qu'un certain seuil noté $BRR_i^{th}(j,t)$, alors le véhicule j est sélectionné pour être le prochain relais. En revanche, si il n'y aucun bon candidat, le voisin qu'est près de la destination avec le maximum $BRR_i(j,t)$ sera choisi.

Le choix de la valeur du seuil :

Le choix est basé sur le rapport calculé de paquet anticipé, $PDR_e(t)$, spécifié par la couche de transport, le protocole BRR pour calculer la valeur de $BRR_i^{th}(j,t)$ comme suit :

$$BRR_i^{th}(j,t) = PDR_e(t)^{D_{ij}/D_{jk}}$$

Où le D_{ij} est la distance entre le véhicule i et J et D_{ik} est la distance entre le véhicule i et le véhicule destination K. En notant que quand le $PDR_e(t) = 0$, les protocoles de BRR et GPSR sont identiques.

2.7.3.2 MCTP

MCTP [8] est un protocole de transport optimisé pour les environnements véhiculaires. Capable de faire la distinction entre les erreurs de liens et les pertes dûes aux congestions du réseau afin de gérer les pertes de paquets de manière appropriée. Il a été développé pour les communications entre les véhicules et un proxy fixe pour garantir l'accès à Internet. MCTP est développé comme une sous-couche entre TCP et IP (voir figure 2.16).

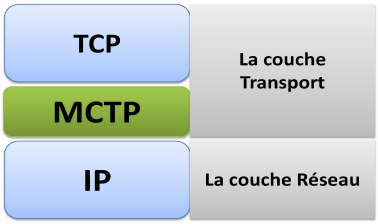

Figure 2.16 – *L'approche cross layer du protocole MCTP.*

MCTP met en œuvre sa propre machine d'état et il peut prendre un des états suivants (voir figure 2.17).

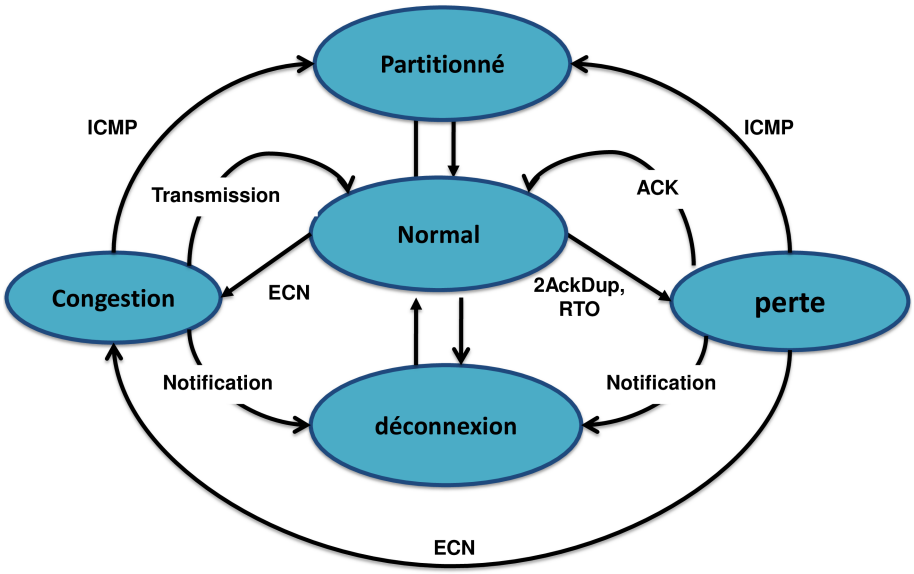

Figure 2.17 – *La machine d'état du protocole MCTP.*

- **État perte** : Si MCTP reçoit 2 (ACK dupliqués) DupAcks ou en cas de l'expiration de RTO, MCTP passe à l'état PERTE et gèle le TCP. Ensuite, MCTP retransmet le segment TCP perdu. Si un ACK est reçu, MCTP passe l'ACK à TCP et TCP le récupère et revient à l'état normal.

- **État congestionné** : Quand ECN indique une congestion venue des nœuds intermédiaires, MCTP passe à l'état congestionné et laisse la situation à TCP pour la gérer très efficacement. Après que le TCP émetteur envoie un nouveau segment, MCTP revient à l'état normal.

- **État partitionné** : Un nœud intermédiaire indique ICMP (destination inaccessible) quand il détecte une déconnexion. MCTP reçoit ce ICMP et se déplace vers l'état

partitionné et gèle TCP. Ensuite il écoute la connexion avec une période constante (non exponentielle). Si MCTP reçoit 1 DupAck, il le récupère par TCP et active le démarrage lent, sans réduire le seuil.

– **État Déconnecté** : Si un véhicule perd le contact à une passerelle, MCTP est avisé et il bascule vers l'état déconnecté dans laquelle il stoppe la transmission TCP et gèle le timer RTO. TCP et MCTP reste dans cet état jusqu'à ce que MCTP est informé de la disponibilité d'une nouvelle passerelle. Puis MCTP restaure TCP et se bascule à l'état normal. Encore une fois, MCTP active la phase de démarrage lent, sans modifier le seuil.

2.7.3.3 VTP (Vanet Transport Control Protocol)

Les principales caractéristiques de la conception VTP [9] sont :

– VTP découple le contrôle de congestion et le contrôle de flux, principalement pour éviter une réduction débit pour la perte de paquets dûe a la non-congestion. En VANETs, les pertes de paquets sont fréquentes en raison de la grande mobilité et les changements topologiques résultants. Ces pertes ne doivent pas invoquer le contrôle de congestion.

– VTP utilise une signalisation explicite de la bande passante disponible de nœuds intermédiaires pour le contrôle de congestion. L'estimation de la bande passante disponible par des nœuds intermédiaires utilise les informations de protocole de couche MAC.

– VTP offre une fiabilité via les retransmissions des paquets perdus. Les accusés de réception sélectifs (SACKs) signalent la perte de paquets à l'expéditeur VTP. Les récepteurs transmettent des SACKs dans des intervalles dynamiques. Ils ajustent l'intervalle selon le taux de transmission en cours et la distance source-destination.

– L'expéditeur VTP utilise les connaissances statistiques pour prédire le comportement de communication attendue d'une connexion. En l'absence d'accusés de réception, la durée de communication prévue pour la distance source-destination assiste le calcul de la temporisation.

Initialement VTP peut être soit dans un état connecté ou perturbé. L'arrivée des accusés de réception (ACK) indique un état connecté. En l'absence d'accusés de réception, l'expéditeur calcule la durée restante prévue de connectivité, en utilisant les résultats statistiques établis précédemment pour la distance source-destination déterminée.

Si le résultat est inférieur à un seuil, VTP passe à l'état perturbé. L'arrivée d'un ACK déclenche la transition de l'état perturbé de l'état connecté.

L'expéditeur VTP continue son transmission au débit de données maximum autorisé, comme un ACK contient la largeur de bande disponible minimale le long du chemin.

Dans un état connecté, un expéditeur VTP utilise la notification explicite, collectés par des nœuds intermédiaires le long du chemin à sauts multiples pour adapter son débit de

données pour les caractéristiques de trajet courant.

Dans un état perturbé, un expéditeur sonde périodiquement si la connectivité est établie. L'intervalle de sondage provient des résultats statistiques sur la base des distances source-destination.

Le contrôle de congestion dans VTP utilise les informations signalé explicitement de nœuds intermédiaires. Le découplage de l'erreur de lien et contrôle de congestion évite les réductions inutiles de taux de transmission dans le cas de la perte de paquets dues à la non-congestion, telles que les erreurs de routage ou des effets sans fil.

Les paramètres d'évaluation du protocole VTP qui caractérisent le chemin de communication sont.

- Une période de connectivité dénote l'existence d'un chemin de bout en bout entre la source et la destination qui permet la communication. Une période de perturbation dénote l'absence d'une telle voie. La durée de connexion décrit par conséquent la longueur d'une période de connectivité alors que la durée de la perturbation décrit la longueur d'une période d'interruption. A noter que des perturbations ne sont pas considérés comme infinis.

- La probabilité de perte de paquets décrit la probabilité qu'un paquet particulier est perdu entre la source et la destination, indépendamment d'autres paquets.

- Le temps d'aller retour (RTT) et RTT gigue. (i) Le RTT décrit le temps entre la transmission d'un paquet et la réception du premier accusé de réception correspondant. La simulation prend un échantillon RTT à un moment donné. (ii) La gigue RTT décrit la différence entre deux échantillons RTT ultérieures. La RTT moyen décrit le moyen pour tous les échantillons de RTT pour une communication donnée.

- La période de remise en ordre décrit le temps à partir de la réception du premier paquet jusqu'à ce que l'arrivée du premier paquet réordonné prévu. Les paquets perdus et dupliqués ne contribuent pas à la réorganisation.

2.8 Tableau récapitulatif

Les protocoles de transport peuvent être classés selon plusieurs critères. On trouve des protocoles passer par l'intermédiaire d'une infrastructure alors que d'autres ne nécessitent pas une infrastructure lors de l'établissement d'une communication. De plus, un autre critère qui a une grande influence sur les performances des protocoles, c'est les mécanismes de détection des pertes.

Dans ce tableau 2.3 on fait le point sur la comparaison des différentes approches.

TABLE 2.3 – *Comparaison*

	I-TCP [4]	**ADTCP** [34]	**ATCP** [3]	**SNOOP** [32]	**ELFN** [6]	**TCP-F** [33]	**VanetTCP** [7]	**MCTP** [8]	**VTP** [9]
Type de solution	Split	cross-layer	cross-layer	Split	cross-layer	cross-layer	cross-layer	cross-layer	cross-layer
La perte rafale	√	√	√	√	×	×			
Détection de déconnexion (Rout failure)	/	STT	ICMP	/	Paquet ELFN	Paquet RFN	RTO	ICMP	Estimation
Mécanisme de reconstruction de Route	/	Sondage (Probing)	Sondage (Probing)	/	Sondage (Probing)	Paquet RRN	Probing	Probing	/
Paquet hors ordre	√	√	√	×	×	×	√	√	√
La sémantique de bout en bout	×	√	√	×	√	√	√	√	√

2.9 SYNTHÈSE ET DISCUSSION

A partir de ce tableau comparatif 2.4 qui résume et illustre clairement les différences entre les deux protocoles TCP pour les Vanets (VanetTcp,MCTP) on conclut des remarques et à partir de ces remarques on arrive à proposé des améliorations.

La première remarque c'est que l'expiration des RTO peut refléter deux choses, ou bien déconnexion ou bien erreur de lien dans la solution VANetTCP l'expiration du RTO signifie une déconnexion, par contre dans la solution MCTP le même évènement (expiration du RTO) interprété par une erreur de lien. Donc pour notre solution **il faut bien préciser la cause de l'expiration du RTO** afin de prendre les meilleures décisions.

La deuxième remarque c'est que l'utilisation du BRR a augmenté considérablement les performances du protocole VanetTCP, pour cela on déduit que **le choix d'un protocole de routage performant est très important**.

TABLE 2.4 – *Comparaison*

	VanetTCP	**MCTP**
Routage	BRR	/
Approche Cross-layer	Top-Down+Sub-layer	Sub-layer
Utilisation des RSU	Non	Oui
Déconnexions	RTO	ICMP
Reconstruction de Route	Probing	Probing
Congestion	ECN	ECN
Perte	DupAck	RTO /DupAck

Conclusion

Dans le présent chapitre, un aperçu général a été présenté sur les différentes techniques d'amélioration du performance de protocole de transport utilisées dans les différents types de réseaux notamment les réseaux sans fil et Adhoc.

Nous avons présenté aussi quelques solutions dans le contexte des réseaux VANets ces solutions permettent de palier les limitations pour améliorer les performances des protocoles TCP pour lesquels les conditions du canal radio varient considérablement comparées à celles d'autre type de réseau.

Dans l'objectif de l'amélioration du rendement de protocole de transport dans les réseaux VANets, nous essayons de contribuer par une nouvelle approche, dans le chapitre suivant, on va illustrée et détaillé.

Notre protocole de Transport 3

Sommaire

Ce chapitre est dédié à la description de la nouvelle solution proposée dans le cadre de protocole TCP dans les réseaux VANets, et fait l'objet également d'une comparaison entre les améliorations déjà proposées précédemment et le nouvel algorithme.

3.1 Introduction

Les améliorations du protocole TCP présentées dans le chapitre précédent ne répondent pas parfaitement aux contraintes imposées par l'environnement VANet et vu la nécessité d'utilisation du protocole TCP dans cette catégorie de réseau (i.e. réseaux VANets), les recherches portant sur l'amélioration du protocole TCP dans cet environnement ne cesse de croitre. La majorité des solutions (i.e. TCP-F, TCP-ELFN, ATCP et TCP-BuS) présentées précédemment pour les réseaux sans fil ad hoc sont basées sur les réactions avec les couches inférieurs (généralement la couche réseau) de la pile protocolaire pour différencier entre la perte des paquets due à la congestion du réseau et la perte des paquets due à un lien erroné.

Malgré la réponse de ces solutions au problème du contrôle de congestion dans les réseaux sans fil ad hoc, certaines d'entre elles ne répondent pas suffisamment au problème de contrôle de congestion du réseau et d'autres créent des conflits entre les communications du protocole TCP standard et les communications du protocole amélioré.

Dans ce chapitre, on présente une nouvelle proposition (solution) pour mieux adapter le protocole TCP aux réseaux VANets. Cette solution se base aussi sur l'interaction des couches inférieures (Réseau et MAC) pour différencier entre les paquets perdus à cause de la congestion et les paquets perdus à cause d'un lien erroné.

Pour éviter les problèmes de conflit dans les différents modèles de perte (congestion, lien erroné, déconnexion) de TCP, notre solution apporte des changements dans les mécanismes de détection du protocole TCP et des changements dans l'agent de routage pour satisfaire le délai de bout en bout de la connexion. En plus, cette solution permet d'éviter la défaillance d'un lien et permet ainsi de minimiser la perte des paquets due à cette coupure.

Les sections qui suivent présentent une description détaillée à l'aide de schémas et d'organigramme de la solution proposée pour améliorer le protocole TCP dans les réseaux VANets.

3.2 Le protocole CVTCP (Cross-layer Vanet Transport Control Protocol)

Notre solution a été proposée pour remédier aux insuffisances des solutions précédentes dans le but d'améliorer les performances du protocole TCP dans les réseaux VANet. Cette solution utilise le même mécanisme de la solution ADHoc tcp ,VANetTCP et MCTP pour différencier entre la perte des paquets due à la congestion du réseau et la perte des paquets due aux échecs des liens.

3.2.1 Principe général

L'idée de notre solution pour remédier à ces problèmes est basée sur les réactions avec les couches inférieures de la pile protocolaire (couche réseau et couche MAC). En conséquence, quelques modifications sont apportées sur le TCP. Pour minimiser la perte des paquets, notre solution s'inspire des solutions précédentes en ajoutant des mécanismes de signalisation dans le nœud intermédiaire ayant détecté un lien erroné, assimilé par l'ajout

de deux métriques SNR et STT pour bien préciser les causes de perte des paquets et pour permettre de choisir les meilleures décisions en plus, on a utilisé une version modifier du protocole de routage nommée EBGR qui est un protocole très meilleur en terme de délai afin d'augmenter les performances du CVTCP.

Cette solution utilise les informations de la puissance du signal des paquets émis pour prédire l'échec des liens et pour minimiser la perte des paquets. Pour éviter la consommation inutile de la bande passante comme dans la solution TCP-ELFN, notre solution garde le même mécanisme de la solution ADHOC-TCP dans le cas d'une découverte d'un nouveau chemin.

Short-Term Throughput (STT) cette métrique a été définie par Fu et al.[34], c'est le débit observé sur un intervalle de temps T.
Dans [34], T est réglé sur $RTT/2$. Le principal avantage de STT est que STT est légèrement affectée par les changements de route transitoires induits par la mobilité dans les MANETs. La congestion du réseau est identifiée par comparaison de la valeur STT à un seuil calculé à partir de tous les échantillons. Cependant, puisque le débit est une valeur absolue qui dépend en grande partie du taux d'envoi, les déconnexions du réseau et les erreurs de canal en rafale, STT ne peut pas fournir la bonne information suffisante pour l'expéditeur ou le destinataire pour distinguer si le réseau est en congestion ou pas.

Signal to Noise Ratio (SNR) défini comme le rapport de la puissance de signal à la puissance du bruit mesuré en dB. Il est utilisé pour indiquer la différence entre le niveau du signal souhaité et de son bruit correspondant.

3.2.2 L'approche cross-layer proposée

Dans notre approche, on a proposé un sublayer CVTCP avec une information cross-layer (bottom-up) des couches inférieures (voir figure 3.1).

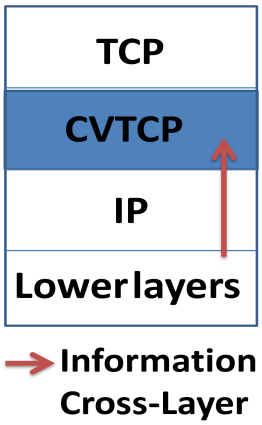

Figure 3.1 – *l'approche cross-layer proposée.*

3.2.3 Protocole de routage

La problématique se focalise spécialement sur l'instabilité des chemins de communication causée par la forte mobilité et le changement de direction fréquent des véhicules qui nous mènent vers le problème de la fragmentation et la déconnexion fréquente.

Le protocole EBGR (Edge node Based Greedy Routing)

C'est un protocole de routage basé sur les positions géographiques destiné aux environnements autoroutières [37]. Le processus d'acheminement des paquets de données est effectué en combinant la stratégie du Greedy Forwarding et la sélection des véhicules relais. En effet, EBGR sélectionne des véhicules situés sur les frontières de la portée de transmission pour acheminer les paquets de données tout en prenant considérations des trajectoires des véhicules qui vont vers la direction de la destination.

La sélection des véhicules voisins (NNI : Neighbor Node Identification)

L'identification des véhicules voisins est réalisée à l'aide de l'échange périodique des messages Hello. Les messages Hello se composent principalement de l'identifiant du véhicule, sa localisation ainsi que le l'instant d'envoi.

Calcul de la distance (DC : Distance Calculation)

Le nœud le plus proche de la destination est identifié à l'aide de cette formule.

$$DC = (1 - D_i/D_c) \tag{3.1}$$

Où
D_i : La plus courte distance entre le nœud i porteur du paquet et la destination D.
D_c : La plus courte distance entre le nœud voisin c et la destination D.

L'identification de la direction de mouvement (DMI : Direction of Motion Identification)

C'est le cosinus d'angle entre le vecteur de vitesse et le vecteur de localisation.

$$DMI = cos(\overrightarrow{V_l}, \overrightarrow{L_{l,d}}) \tag{3.2}$$

Où
$\overrightarrow{V_i}$: Vecteur de vitesse du véhicule de bord i.
$\overrightarrow{L_{i,d}}$: Vecteur de localisation du véhicule de bord i.

Calcul de la stabilité du lien (LS : Link Stability)

La stabilité du lien entre deux nœuds pendant une période de temps T est déterminée comme étant le rapport entre le Maximum Transmission Range et la distance entre ces deux nœuds pendant T.

$d_1 = v_1 t$

$d_2 = v_2 t$

$$x_1 = \grave{x}_1 + x_1 = \grave{x}_1 + d_1 \cos(\theta_1) = \grave{x}_1 + t(v_1 \cos(\theta_1))$$

$$y_1 = \grave{y}_1 + y_1 = \grave{y}_1 + d_1 \sin(\theta_1) = \grave{y}_1 + t(v_1 \sin(\theta_1))$$

$$x_2 = \grave{x}_2 + x_2 = \grave{x}_2 + d_2 \cos(\theta_2) = \grave{x}_2 + t(v_2 \cos(\theta_2))$$

$$y_2 = \grave{y}_2 + y_2 = \grave{y}_2 + d_2 \sin(\theta_2) = \grave{y}_2 + t(v_2 \sin(\theta_2))$$

$$D^2 = \{(\grave{x}_1 - \grave{x}_2) + t(v_1 \cos(\theta_1)\}^2 + \{(\grave{y}_1 - \grave{y}_2) + t(v_1 sin(\theta_1)\}^2$$

$$D = \sqrt{\{(\grave{x}_1 - \grave{x}_2) + t(v_1 \cos(\theta_1)\}^2 + \{(\grave{y}_1 - \grave{y}_2) + t(v_1 sin(\theta_1)\}^2}$$

$$LS = \frac{R}{D} = \frac{R}{\sqrt{\{(\grave{x}_1 - \grave{x}_2) + t(v_1 \cos(\theta_1)\}^2 + \{(\grave{y}_1 - \grave{y}_2) + t(v_1 sin(\theta_1)\}^2}} \tag{3.3}$$

Calcul du score potentiel (PS : Potontial Score Calculation)

Un score potentiel est calculé pour tous les nœuds présent dans la portée de transmission (LTR)(level transmission range) du véhicule source. Ce score est calculé afin d'identifier la trajectoire des véhicules. Le véhicule avec le score potentiel le plus élevé sera considéré comme le véhicule qui va vers la direction de la destination D, et sera par conséquent choisi pour être le prochain véhicule relai pour acheminer le paquet de données jusqu'au véhicule destination D. Le score potentiel est calculé en se basant sur la formule suivante :

$$PS_i = p \times (1 - \frac{D_i}{D_c}) + w \times cos(\overrightarrow{V_l}, \overrightarrow{L_{l,d}}) + \beta \times LS_{c,i} \tag{3.4}$$

Où
PS_i : Score potentiel du nœud i.
p, w : Deux facteurs potentiels, où $p + w + \beta = 1$ et $\beta > p$ et $\beta > w$.
D_i : La plus courte distance entre le nœud i et la destination D.
D_c : La plus courte distance entre le nœud courant c et la destination D.
$\overrightarrow{V_i}$: Vecteur de vitesse du véhicule de bord i.
$\overrightarrow{L_{i,d}}$: Vecteur de localisation du véhicule de bord i.

La sélection des véhicules de bord (ENS : Edge Node Selection)

Ce processus consiste à sélectionner le véhicule le plus proche de la destination se trouvant dans la portée de transmission du véhicule source. La portée de transmission du nœud porteur du paquet est virtuellement divisée en cinq différents niveaux en fonction de la distance. Les nœuds voisins se trouvent a une distance entre 250 et 200 m seront favorisés par rapport au nœuds voisins situés a une la distance entre 200 et 150 m.

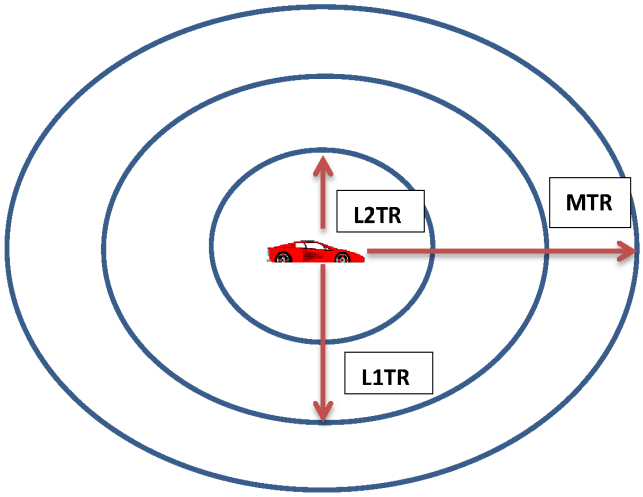

Figure 3.2 – *les différents niveaux de portée de transmission.*

Dans notre cas, on a décidé de divisé la portée en trois sous-portées virtuelles de 100m. Voir la figure 3.2 et le calcul de score se fait en utilisant l'équation suivante :

$$PS_i = (1 - \frac{D_i}{D_c}) + cos(\overrightarrow{V_l}, \overrightarrow{L_{l,d}}) + LS_{c,i} \tag{3.5}$$

L'algorithme 1 récapitule le mécanisme de fonctionnement du protocole EBGR :

Algorithme 1 : pseudo code d'EBGR

Variables :

C : *Véhicule courant*;
D : *Véhicule destination*;
Loc_i : *Position du véhicule* $\boldsymbol{i}$;
N_i : *Le* $\boldsymbol{i}^{ème}$ *voisin*;

Début
  $ProchainSaut \leftarrow C$;
  pour tous les N_i *de* C **faire**
    $D_i \leftarrow distance(Loc_D, Loc_i)$;
    $D_{ci} \leftarrow distance(Loc_C, Loc_i)$;
    pour tous les N_i *de* C **faire**
      Si $(D_{ci} < MTR)$ *and* $(D_{ci} > L1TR)$ **Alors**
        $PS_i = p \times (1 - \frac{D_i}{D_c}) + w \times cos(\overrightarrow{V_l}, \overrightarrow{L_{l,d}})$;
        pour tous les N_i *avec un grand* PS_i **faire**
          PS=PS_i;
          $ProchainSaut \leftarrow N_i$;
        fin
      Sinon
        Si $(D_{ci} < L1TR)$ *and* $(D_{ci} > L2TR)$ **Alors**
          $PS_i = p \times (1 - \frac{D_i}{D_c}) + w \times cos(\overrightarrow{V_l}, \overrightarrow{L_{l,d}})$;
          pour tous les N_i *avec un grand* PS_i **faire**
            PS=PS_i;
            $ProchainSaut \leftarrow N_i$;
          fin
        Sinon
          Si $(D_{ci} < L2TR)$ **Alors**
            $PS_i = p \times (1 - \frac{D_i}{D_c}) + w \times cos(\overrightarrow{V_l}, \overrightarrow{L_{l,d}})$;
            pour tous les N_i *avec un grand* PS_i **faire**
              PS=PS_i;
              $ProchainSaut \leftarrow N_i$;
            fin
          Sinon
            Carry & forward (Paquet, C) ;
          Finsi
        Finsi
      Finsi
    fin
  fin
  Si $ProchainSaut \neq C$ **Alors**
    Envoyer (Paquet, ProchainSaut) ;
  Sinon
    Carry & forward (Paquet, C) ;
  Finsi
Fin

3.2.4 La Machine à États

CVTCP met en œuvre sa propre machine à états (voir la figure 3.3). Il peut prendre un des états suivants :

3.2.4.1 État normal

Dans cet état CVTCP écoute les informations de l'état du réseau fournies par le message explicite de notification de congestion ECN et par le message ICMP « des destinations inaccessibles ». Selon l'information reçue le CVTCP bascule vers l'état approprié.

3.2.4.2 État congestion

Dans les réseaux câblés, une source est implicitement informée de la congestion du réseau quand la retransmission time-out (RTO) expire ou trois ACK dupliqués sont reçus. Une étude précédente [38],[39] propose le mécanisme de la notification explicite de congestion (ECN), dans lequel les routeurs dans le chemin vers la destination explicitement informent la source d'empêcher la congestion du réseau.

Lorsque l'un des routeurs le long du chemin vers la destination détecte que la charge de la file d'attente a dépassé un seuil donné, il envoie un message ECN qui définit la congestion Expérimentée (CE) mise dans l'entête des paquets IP pour aviser la source. Le réglage du bit CE est une fonction de probabilité basée sur la taille de la file d'attente. A la réception de ce message d' ECN, la source invoque immédiatement le système de contrôle de congestion à diminuer son taux de transmission sans attendre un événement de temporisation ou trois ACK dupliqués.

L'approche proposée suit le même chemin de la mise en œuvre, en utilisant le mécanisme ECN pour identifier les pertes de paquets en raison de débordement de tampon. Quand un nœud le long du chemin vers la destination détecte que la taille de la mémoire tampon est supérieure à un seuil donné, on positionne le bit ECN de paquets. Si le bit ECN est marqué dans un paquet de données reçu, la destination doit également positionner le bit ECN dans le paquet de retour d'accusé de réception pour notifier la source.

3.2.4.3 État Perte

CVTCP surveille les ACK reçus. Quand CVTCP observe que trois ACK dupliqués ont été reçus ou il y a une expiration de RTO avec une valeur de SNR basse, il va basculer vers l'état de perte, où il doit sauvegarder le dernier nombre de séquence des paquets transmis. Dans cet état, le protocole envoie seulement les segments perdus qui sont indiqués par les ACKs doublés (transmission de toutes les données autorisées par la fenêtre de congestion sans réduction du débit de transmission ou de la taille de la fenêtre de congestion). Le système revient à l'état NORMAL s'il reçoit un nouveau ACK avec un nombre de séquence plus grand que le précédent, attestant que tous les paquets perdus ont été recouvrés.

3.2.4.4 État Déconnexion

À la réception d'un message ICMP « *destination inaccessible* » le système va basculer à l'état de DECONNECTION quelque soit l'état courant, de même si le RTO expire et on a une valeur $STT \simeq 0$. Le système doit sauvegarder les paramètres courants (telle que taille

de fenêtre de congestion, RTT etc...). L'agent TCP dans cet état est gelé et aucun paquet n'est envoyé jusqu'à la découverte d'un nouveau chemin en testant le réseau.

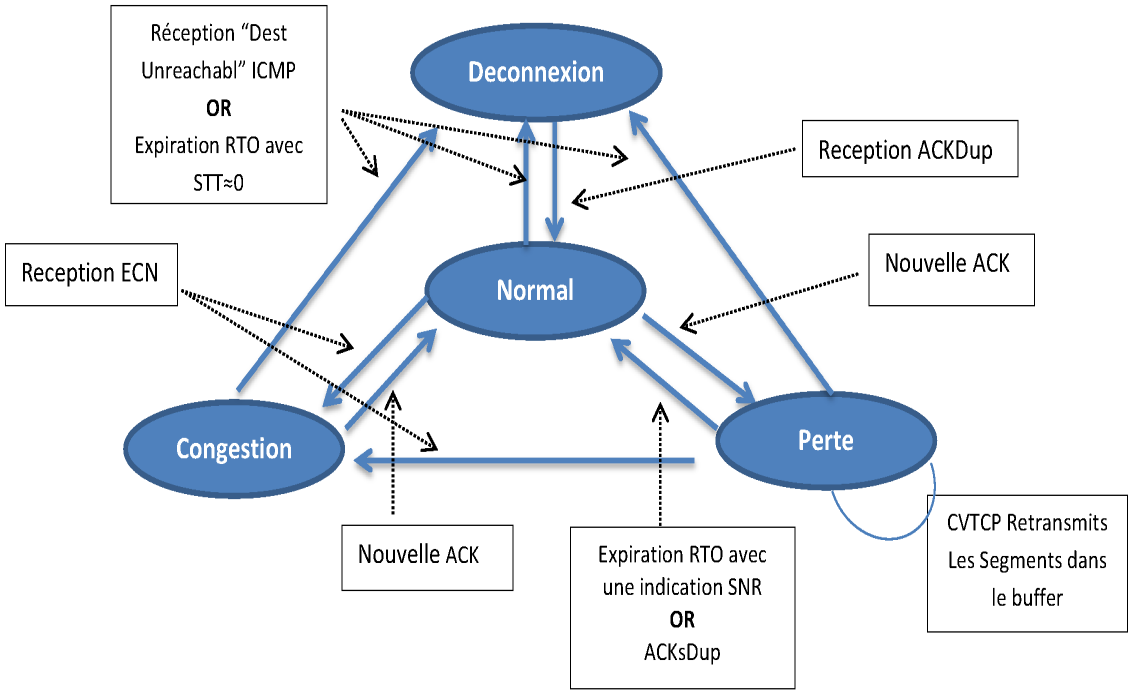

Figure 3.3 – *L'automate CVTCP.*

3.2.5 Les organigrammes du Protocole CVTCP

3.2.5.1 Lors de la réception

Nous avons mis CVTCP comme une couche entre TCP et IP, (voir figure 3.1). CVTCP intercepte tous les paquets allant de la couche IP vers la couche TCP. Il examine les entêtes TCP du paquet et découvre à quelle connexion TCP en examinant l'adresse source du paquet, l'adresse de destination et le numéro de port.

Penchons-nous sur le comportement de CVTCP à l'état normal. Dans cet état, il vérifie d'abord l'ECN. Si le bit ECN est fixé à un (1), l'algorithme définit l'état de CVTCP à congestionné et passe ensuite le segment à TCP. A la réception d'un paquet de données avec ECN=1, TCP exécute l'algorithme de contrôle de congestion car une congestion du trafic réseau a été détectée par un routeur quelque part entre l'émetteur et le récepteur. L'algorithme rend CVTCP à l'état normal de l'état congestionné lorsque TCP envoie un paquet.

Si le drapeau ECN n'est pas défini, CVTCP compte le nombre des ACKs dupliqués reçus et met le TCP en mode "persistent" si il reçoit 3 ACK en double. CVTCP bascule vers l'état de perte et traite le segment. Un autre cas dans lequel CVTCP passe à l'état de perte (et met en TCP persistent) se produit lorsque RTO de TCP est sur le point d'expirer avec une valeur de SNR basse comme le montre l'organigramme 3.4.

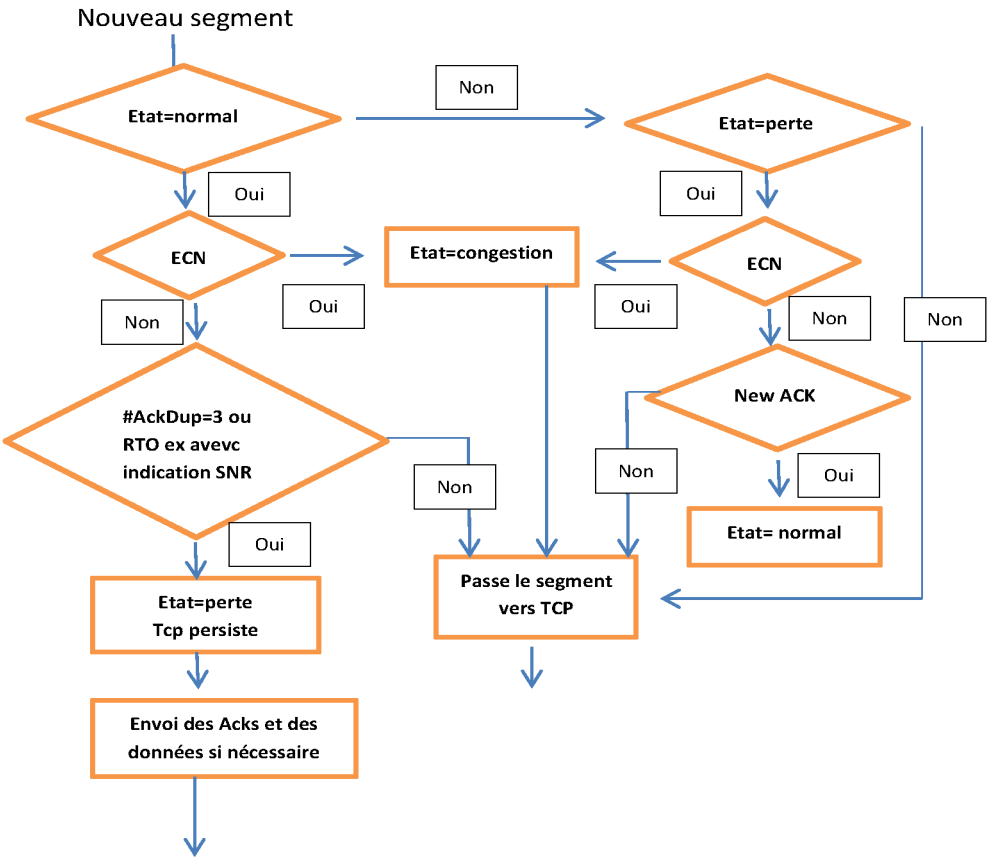

Figure 3.4 – *Comportement de CVTCP en cas de réception.*

3.2.5.2 Lors de l'envoi

Après le traitement nécessaire, les paquets sont envoyés selon l'état approprié du CVTCP vers la couche réseau, voir la figure 3.5.

3.3 Évaluation des performances du CVTCP

Afin d'évaluer les performances de notre protocole de transport, un ensemble de simulations a été effectué, pour pouvoir identifier les facteurs influant sur ses performances. L'objectif est de présenter les résultats de simulation de notre approche, suivie d'une étude comparative de performances avec d'autres protocoles comme (ADTCP [34])(VANetTCP[7]).

3.3.1 Environnement de simulation

Pour étudier et évaluer les performances de notre protocole nous avons eu recours à la simulation, avec l'utilisation de l'outil de simulation NS-2 [40], qui est Open Source disponible sur toutes les plateformes, nous avons également choisi un environnement *freeway* plus proche de la réalité en utilisant le générateur de mobilité IMPORTANT TOOL [41], afin de générer les fichiers traces définissant des scénarios de mobilité. Cet outil a été proposé dans une université américaine afin que les simulations liées aux réseaux véhiculaires soient le plus réaliste, le tout installé et implémenté sous UBUNTU 10.10.

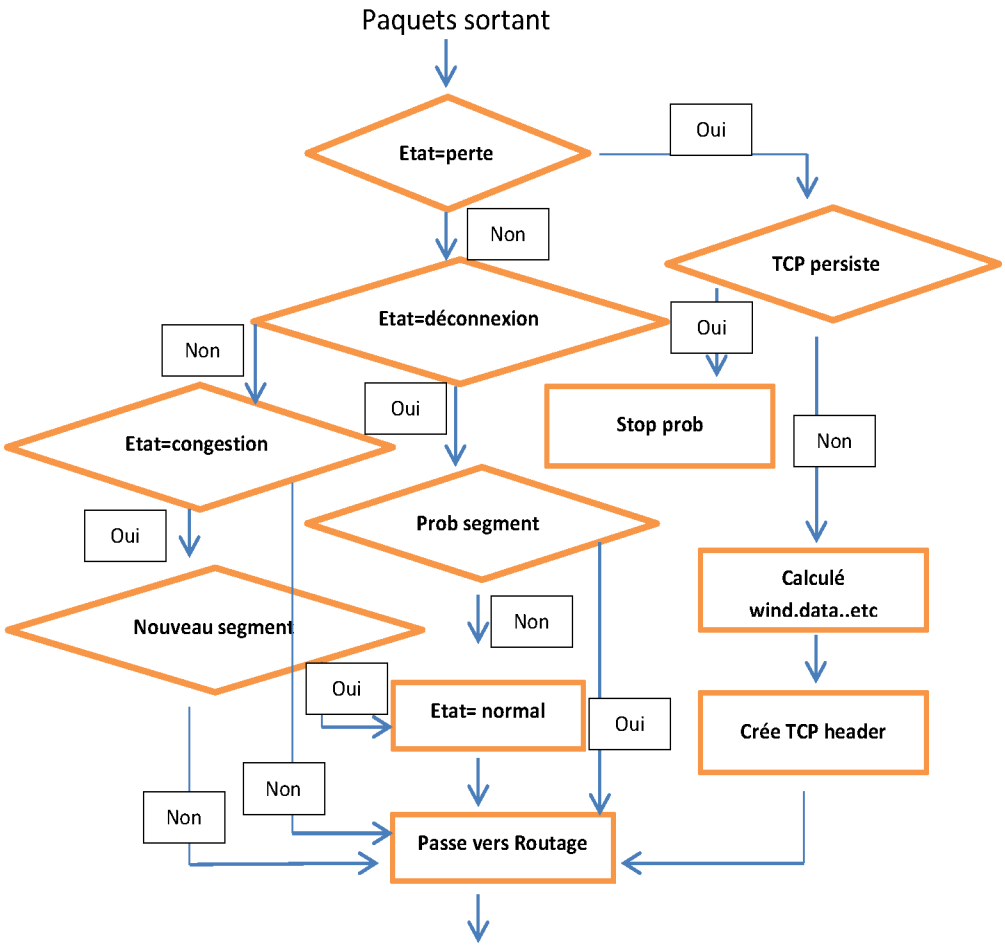

Figure 3.5 – *Comportement de CVTCP en cas d'envoi.*

3.3.2 Paramètres de simulation

Pour réaliser notre simulation, nous avons défini plusieurs paramètres, certains paramètres sont fixés et ne changent pas durant toute la simulation, d'autres sont variables pour permettre d'obtenir à chaque fois un nouveau scénario de simulation. La variation de ces paramètres nous permet d'identifier ceux ayant une influence sur la performance de notre approche. Les scénarios de simulation sont exécutés sur une route droite de 10 Km, composée de 2 voies bidirectionnelles. Chaque nœud dans la simulation a une portée de 300 m et se déplace d'une moyenne de 50 à 150 Km/h. On sélectionne a chaque scénario entre 5 et 50 pairs effectuant des échanges de paquets de données d'une taille de 1 Kbyte.

On a varié le nombre de nœuds « 20, 40, . . ., 200 ». Le tableau 3.1 ci-dessous récapitule l'ensemble des paramètres de simulations utilisés.

3.3.3 Le choix des métriques appropriées pour la comparaison

Les métriques utilisées pour mesurer les performances du protocole dans ce mémoire sont les suivantes :

Débit (THROUGHPUT) :est la quantité de données qu'un véhicule reçoit de l'expédi-

Scénario de		Simulation	
Temps de simulation	10 secondes	Protocole MAC	802.11
langueur de la route	10 Km	Modèle de propagation	Two-Ray -ground
Nombre de paires communicants	5 - 50	Portée de communication	300 m
Nombre de voie	2	Taille des paquets	1 KByte
Nombre de véhicules	20 - 200	Modèle de mobilité	Freeway
Vitesse des véhicules	50 - 150 Km/h	Capacité du canal	1 Mbps
Type de trafic	FTP	Protocoles de routage	MyEBGR,DSR,BRR

TABLE 3.1 – *Paramètres de simulation.*

teur dans un intervalle de temps spécifique.

Débit = total des données reçues/la durée totale

La durée totale est la différence entre les moments de réception du dernier paquet et le premier paquet émis.

Délai de bout en bout (EED) : est le temps total pris par un paquet au cours de la transmission dans un réseau d'une entité communicante à l'autre. Le délai est calculé en prenant la différence entre les moments de l'envoi et la réception d'un paquet.

Délai = temps d'arrivée du paquet (A) - temps d'envoi du paquet (A).

3.3.4 Résultats et interprétations

Dans cette section, nous illustrons et nous analysons les résultats de simulation que nous avons obtenu à travers les scénarios de simulation précédemment définis.

Throughput & EED vs Nombre de communication :

La figure 3.6 montre la performance de trois versions différentes (la solution proposée, ADTCP, VANetTCP) en termes de débit en fixant le nombre de nœuds à 100 et en variant a chaque fois le nombre de pairs communicants.

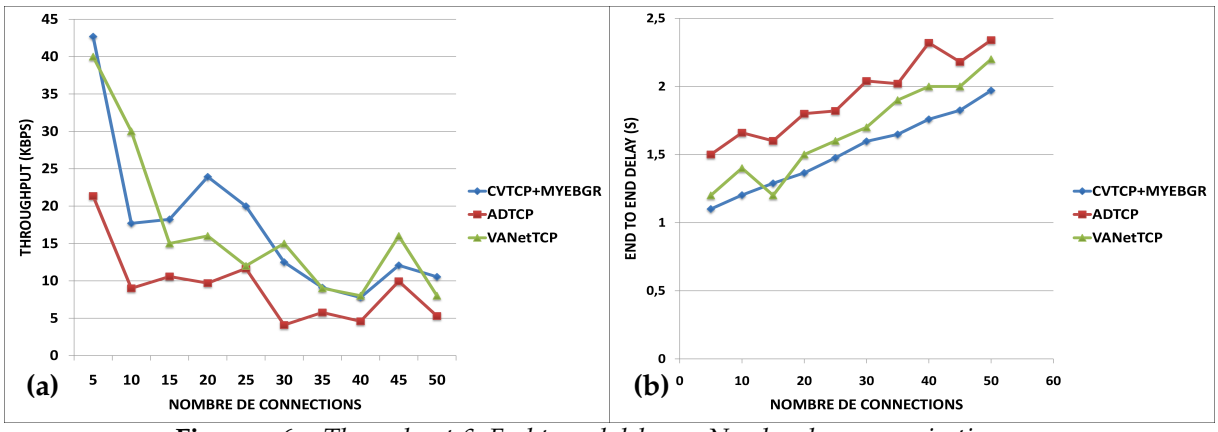

Figure 3.6 – *Throughput & End to end delay vs Nombre de communication.*

D'après les courbes de la figure 3.6 CVTCP offre une performance supérieure par rapport aux autres protocoles, il fournit significativement un meilleur débit que ADTCP et le débit atteint est près de VANetTCP. Cependant, on constate que la performance en terme de throughput se dégrade en augmentant le nombre de connections, cela est justifié par :

- La perte de plusieurs paquets dans le réseau suite à des collisions.
- Les problèmes de nœuds cachés et exposés.

Le protocole ADTCP offre un débit réduit cela est principalement dû à deux raisons :

- l'inefficacité du mécanisme de routage, le chemin choisi par le protocole DSR, une voie extrêmement fragile, donc très peu de paquets sont correctement transmis à la destination sans se faire chuté le long du chemin.
- ADTCP réduit son taux de transmission pour chaque perte de paquet même si la congestion n'est pas la cause de perte. Par conséquent, la taille de la fenêtre de transmission de protocole ADTCP reste généralement très petite. Donc, très peu de paquets sont transmis par le véhicule source au cours de l'ensemble des simulations.

Par contre, notre technique proposée est conçue pour éviter les problèmes mentionnés ci-dessus. Le protocole VANetTCP est affecté par la distance entre la source et la destination, donc il montre un débit passable pour les scénarios où la source et la destination sont à moins de 400m de distance, la plupart des améliorations de débit provient du protocole de routage BRR.

D'après ces résultat, notre solution peut fournir une amélioration significative en termes de débit, puisque les paquets doivent se déplacer le long d'un parcours relativement court et souvent déconnectée. Donc, la déconnexion du réseau devient plus fréquente. Alors en identifiant correctement les causes des pertes de paquets, CVTCP est capable de reprendre après une interruption rapidement et peut atteindre un débit plus élevé .

Concernant le délai de bout en bout, nous voyons bien que la performance atteinte avec notre approche est nettement meilleure à celle des autres protocoles, ceci est grâce à la précision du calcul de la variation du trafic routier dans les segments de route ainsi qu'à l'approche de division virtuelle de la portée en trois petits niveaux successives. En

outre, les protocoles DSR et BRR souffrent bien de délai assez important par rapport à notre approche, cela est dû exclusivement à la découverte des chemins de bout en bout qui engendre des délais supplémentaires.

Tant que dans MyEBGR les chemins sont établis progressivement en prenant en considération la variation temporelle du trafic routier et en choisissant les chemins les plus cours garantis sans un minimum de perte de message (LS), contrairement aux BRR qui prend des chemins sûr mais probablement long, ce qui explique les résultats obtenus. Cependant pour DSR, le mécanisme de découverte préalable des chemins le rend très sensible aux changements fréquents de la topologie.

Throughput & End to end delay vs Nombre de Nœuds :

La figure 3.7 représentent le throughput et le End to End Delay en fonction de la densité du réseau, c'est à- dire le nombre total des véhicules en fixant le nombre des paires communicants à 5.

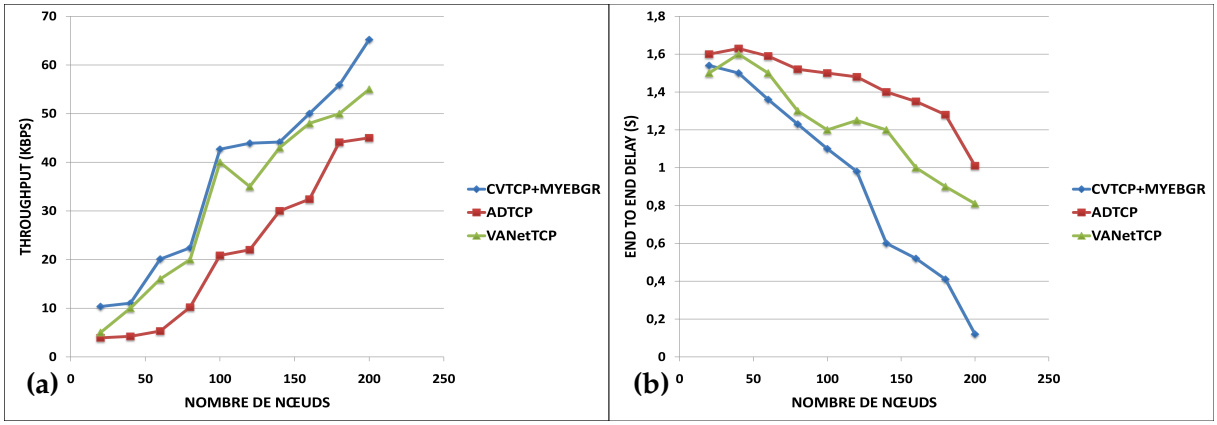

Figure 3.7 – *Throughput & End to end delay vs Nombre de Nœuds.*

La figure 3.7 montre que le throughput de notre approche s'augmente au fur et à mesure de l'ajout des véhicules sur le réseau, c'est un comportement logique puisque le réseau devient de plus en plus connecté donc peu de paquets seront perdus.

Concernant le délai de bout en bout, le résultat est clair, CVTCP montre une diminution après chaque augmentation du nombre de véhicules sur le réseau, cela est dû à l'augmentation de la connectivité, permettant ainsi au Protocole de routage de faire transiter les paquets avec succès à leurs destinations avec un délai réduit. Tandis que les deux autres protocoles, ont un délai de bout en bout qui n'est pas assez réduit, dû à l'utilisation de la phase de découverte et le mécanisme de calcul de next forwarder par les deux approches DSR du ADTCP et BRR du VANetTCP respectivement, qui certainement pénalisent le délai d'envoi des paquets.

Conclusion

Dans ce chapitre on a présenté notre solution pour améliorer les performances du protocole TCP dans les réseaux véhiculaires, cette solution qu'on a baptisée CVTCP tente de remédier aux insuffisances des solutions existantes.

Contrairement aux autres solutions, notre solution utilise les capacités des deux couches inférieurs (Réseau et MAC) pour contrôler réellement la congestion du réseau et utilise la puissance du signal pour réduire la perte des paquets.

Il faut noter que notre solution peut être intégrée avec n'importe quel protocole de routage. Pour mieux mettre en évidence les apports de notre solution, une analyse à base de simulation sous le simulateur NS-2 a été faite.

Conclusion et Perspectives

Ce mémoire s'articule sur deux volets, dont le premier consistait à faire un bilan sur les travaux concernant le protocole TCP dans tous les types de réseaux (filaire, ad hoc, VANet) afin d'identifier leurs manques, et le second, de proposer une nouvelle variante pour le protocole TCP. Nous avons proposé une solution qui tend à améliorer les performances de TCP dont les mesures ont été effectuées par simulation sous Network Simulator.

Notre proposition est basée sur les interactions avec les couches inférieures pour remédier à la dégradation des performances du TCP dans les réseaux VANet. Le changement fréquent de la topologie dû à la mobilité des nœuds, et l'incertitude des liens sans fil ont poussé les chercheurs et les développeurs des réseaux ad hoc à faire face à ces nouveaux défis pour mieux adapter le protocole TCP dans cette catégorie de réseau (i.e. VANet). L'incapacité du protocole TCP traditionnel à distinguer entre les erreurs de transmission causées par des liens erronés, et celles causées par la congestion du réseau influent considérablement sur les performances du protocole TCP traditionnel dans les réseaux VANet. Ce qui mène vers une mauvaise utilisation de la bande passante, et une exécution d'un contrôle de congestion dans des situations inappropriées.

Ces dernières années, peu de solutions ont été proposées, où chacune d'elles tente à améliorer les performances du protocole TCP dans les réseaux VANet. Toutes ces solutions ont un mécanisme commun pour distinguer entre la perte des paquets provoqués par la rupture des liens, et celle due la congestion du réseau. Elles utilisent la réaction avec la couche réseau pour notifier les ruptures des chemins, mais elles ne répondent pas parfaitement aux contraintes imposées par les réseaux sans fil. Pour remédier aux insuffisances de ces solutions, on a proposé une nouvelle variante qu'on a appelé CVTCP.

CVTCP est une amélioration des solutions ATCP, VANetTCP, MCTCP qui repose sur le même principe de distinction entre les causes de la perte de paquets. Elle utilise des tampons pour conserver les paquets qui doivent être acheminés une fois le chemin vers la destination est rétabli. Dans CVTCP, on se base sur la couche MAC pour notifier les liens erronés, ce qui permet de geler rapidement la connexion TCP avant que l'expéditeur transmet des paquets supplémentaires vers la destination. Cette solution utilise seulement un tampon unique au niveau du nœud intermédiaire ayant détecté un lien erroné pour sauvegarder les paquets à acheminer vers la destination. Cette technique apporte un gain considérable en termes de place mémoire.

La validation de la solution dans un réseau VANet réel est une question clé de la recherche. Afin d'éviter cette réalité, on a procédé par simulation. Actuellement, une grande variété de simulateurs existe pour simuler les réseaux informatiques. Pour valider notre solution, on a utilisé le simulateur NS-2, puisqu'il produit des résultats plus proches à la réalité et il supporte l'extension de nouveaux modèles de protocole.

En se basant sur un ensemble d'études, la validation de la solution CVTCP a montré une nette amélioration des performances de ce dernier, en comparaison avec le protocole TCP traditionnel disponible sur le simulateur ns-2. La série de tests effectués a prouvé que

CVTCP répond parfaitement aux phénomènes de congestion du réseau et distingue sans aucune ambigüité entre les deux causes de pertes de paquets.

Notre solution a permis d'ouvrir de nouvelles perspectives, et spécialement l'utilisation d'autres paramètres autrement que la puissance du signal et le STT. Ces perspectives nous orientent a pensé autrement, et à concevoir de nouvelles solutions pour mieux adapter le protocole TCP dans l'environnement mobile ad hoc afin de contrôler la congestion dans les situations appropriées.

Comme autre perspective à notre travail, il est nécessaire de penser à utiliser les RSU et il est utile de s'intéresser aux calculs de paramètres tels que le temps d'aller-retour (RTT), le temps d'expiration de la découverte du chemin ... etc. L'utilisation de ces paramètres d'une manière efficace permettra sûrement de minimiser les délais et aussi d'utiliser la bande passante d'une manière efficace, ce qui contribuera forcement à améliorer les performances du protocole TCP dans les réseaux VANet.

Bibliographie

[1] Ahmed Soua. *Vehicular ad hoc networks : dissemination, data collection and routing : models and algorithms*. PhD thesis, Institut National des Télécommunications, 2013. (Cité pages ix, 11 et 12.)

[2] Kahina Ait Ali. *Modélisation et étude de performances dans les réseaux VANET*. PhD thesis, Belfort-Montbéliard, 2012. (Cité pages ix, 12, 13 et 14.)

[3] Jian Liu and Suresh Singh. Atcp : Tcp for mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 19(7) :1300–1315, 2001. (Cité pages ix, 1, 36, 37, 39 et 45.)

[4] Ajay Bakre and BR Badrinath. I-tcp : Indirect tcp for mobile hosts. In *Distributed Computing Systems, 1995., Proceedings of the 15th International Conference on*, pages 136–143. IEEE, 1995. (Cité pages 1, 31 et 45.)

[5] Andrea Zanella, Gregorio Procissi, Mario Gerla, and MY Sanadidi. Tcp westwood : Analytic model and performance evaluation. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 3, pages 1703–1707. IEEE, 2001. (Cité page 1.)

[6] Van Jacobson. Modified tcp congestion avoidance algorithm. *end2end-interest mailing list*, 1990. (Cité pages 1, 35 et 45.)

[7] Wantanee Viriyasitavat, Fan Bai, and Ozan K Tonguz. Toward end-to-end control in vanets. In *Vehicular Networking Conference (VNC), 2011 IEEE*, pages 78–85. IEEE, 2011. (Cité pages 2, 39, 45 et 56.)

[8] Marc Bechler, Sven Jaap, and Lars Wolf. An optimized tcp for internet access of vehicular ad hoc networks. In *NETWORKING 2005. Networking Technologies, Services, and Protocols ; Performance of Computer and Communication Networks ; Mobile and Wireless Communications Systems*, pages 869–880. Springer, 2005. (Cité pages 2, 42 et 45.)

[9] R Schmilz, Alain Leiggener, Andreas Festag, Lars Eggert, and Wolfgang Effelsberg. Analysis of path characteristics and transport protocol design in vehicular ad hoc networks. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 2, pages 528–532. IEEE, 2006. (Cité pages 2, 43 et 45.)

[10] Sofiane Khalfallah, Moez Jerbi, Mohamed Oussama Cherif, Sidi-Mohammed Senouci, Bertrand Ducourthial Ducourthial, et al. Expérimentations des communications inter-véhicules. In *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP)*, 2008. (Cité page 6.)

[11] Nouha BACCOUR SELLAMI. *Master's thesis : conception d'une nouvelle stratégie de routage dynamique pour les réseaux mobiles ad hoc*. PhD thesis, Université de Sfax, 2006. (Cité page 6.)

[12] Noureddine CHAIB. *Mémoire de magister : La sécurité des communications dans les réseaux vanet.* PhD thesis, Université elhadj lakhder BATNA, 2011. (Cité pages 7 et 8.)

[13] Amadou Adama Ba. Protocole de routage basé sur des passerelles mobiles pour un accès internet dans les réseaux véhiculaires. 2011. (Cité page 7.)

[14] SP Vaishali D Khairnar and SN Pradhan. V2v communication survey-(wireless technology). *International Journal of Computer Technology & Applications*, 3(1) :370–373, 2012. (Cité page 7.)

[15] Oscar Trullols Cruces. Applying delay tolerant protocols to vanets. In *Univ. of Catalonia*, 2008. (Cité page 8.)

[16] CAMP Vehicle Safety Communications Consortium et al. Vehicle safety communications project : task 3 final report : identify intelligent vehicle safety applications enabled by dsrc. *National Highway Traffic Safety Administration, US Department of Transportation, Washington DC*, 2005. (Cité page 8.)

[17] Jonathan Petit. *Surcoût de l'authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2011. (Cité pages 8, 10, 14 et 15.)

[18] Daniel Jiang and Luca Delgrossi. Ieee 802.11 p : Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040. IEEE, 2008. (Cité page 12.)

[19] Andreas Festag and Soeren Hess. Etsi technical committee its : news from european standardization for intelligent transport systems (its)-[global communications newsletter]. *Communications Magazine, IEEE*, 47(6) :1–4, 2009. (Cité page 13.)

[20] Ismail Djama. *Adaptations inter-couches pour la diffusion des services vidéo sans fil*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 2008. (Cité pages 15 et 16.)

[21] Vineet Srivastava and Mehul Motani. Cross-layer design : a survey and the road ahead. *Communications Magazine, IEEE*, 43(12) :112–119, 2005. (Cité pages 16 et 18.)

[22] Mihaela van Der Schaar et al. Cross-layer wireless multimedia transmission : challenges, principles, and new paradigms. *Wireless Communications, IEEE*, 12(4) :50–58, 2005. (Cité pages 16 et 18.)

[23] Frédéric Nivor. *Architecture de communication pour les applications multimédia interactives dans les réseaux sans fil*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2009. (Cité page 17.)

[24] Sakuna Charoenpanyasak. *Optimisation inter-couches du protocole SCTP en réseaux ad hoc*. PhD thesis, 2008. (Cité pages 22, 28 et 38.)

[25] Savo Glisic and Beatriz Lorenzo. *Advanced wireless networks : cognitive, cooperative & opportunistic 4G technology*. John Wiley & Sons, 2009. (Cité page 24.)

[26] Douglas Comer. *Internetworking with TCP/IP, Vol. I : Principles, Protocols, and Architecture, 3/e*. Englewood Cliffs (NJ) : Prentice-Hall, 1995. (Cité page 25.)

[27] Laâredj Chellama. *Gestion de la mobilite au niveau de la couche transport*. PhD thesis, 2013. (Cité pages 25, 31, 32, 34 et 36.)

[28] BN Yuvaraju and Niranjan N Chiplunkar. Scenario based performance analysis of variants of tcp using ns2-simulator. *International Journal of Advancements in Technology*, 1(2) :223–233, 2010. (Cité page 27.)

[29] Van Jacobson. Congestion avoidance and control. In *ACM SIGCOMM Computer Communication Review*, volume 18, pages 314–329. ACM, 1988. (Cité page 27.)

[30] Hala Elaarag. Improving tcp performance over mobile networks. *ACM Computing Surveys (CSUR)*, 34(3) :357–374, 2002. (Cité page 31.)

[31] Kevin Brown and Suresh Singh. M-tcp : Tcp for mobile cellular networks. *ACM SIGCOMM Computer Communication Review*, 27(5) :19–43, 1997. (Cité page 32.)

[32] Hari Balakrishnan, Srinivasan Seshan, and Randy H Katz. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 1(4) :469–481, 1995. (Cité pages 33 et 45.)

[33] Kartik Chandran, Sudarshan Raghunathan, Subbarayan Venkatesan, and Ravi Prakash. A feedback-based scheme for improving tcp performance in ad hoc wireless networks. *Personal Communications, IEEE*, 8(1) :34–39, 2001. (Cité pages 35 et 45.)

[34] Zhenghua Fu, Benjamin Greenstein, Xiaoqiao Meng, and Songwu Lu. Design and implementation of a tcp-friendly transport protocol for ad hoc wireless networks. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 216–225. IEEE, 2002. (Cité pages 37, 45, 49 et 56.)

[35] Thomas D Dyer and Rajendra V Boppana. A comparison of tcp performance over three routing protocols for mobile ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 56–66. ACM, 2001. (Cité page 38.)

[36] Feng Wang and Yongguang Zhang. Improving tcp performance over mobile ad-hoc networks with out-of-order detection and response. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 217–225. ACM, 2002. (Cité page 38.)

[37] K Prasanth, K Duraiswamy, K Jayasudha, and C Chandrasekar. Minimizing end-to-end delay in vehicular ad hoc network using edge node based greedy routing. In *Advanced Computing, 2009. ICAC 2009. First International Conference on*, pages 135–140. IEEE, 2009. (Cité page 50.)

[38] Kadangode Ramakrishnan and Sally Floyd. A proposal to add explicit congestion notification (ecn) to ip. Technical report, RFC 2481, January, 1999. (Cité page 54.)

[39] K Ramakrishnan, Sally Floyd, David Black, et al. The addition of explicit congestion notification (ecn) to ip, 2001. (Cité page 54.)

[40] Network Simulator. Network simulator 2. 2013. (Cité page 56.)

[41] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. User manual for important mobility tool generators in ns-2 simulator. *University of Southern California*, 2004. (Cité page 56.)

Liste des abréviations

ACK	Acknowlegment
Airmail	Asymmetric reliable mobile access
AODV	Ad hoc On-Demand Distance Vector
ATCP	Ad hoc Transmission Control Protocol
BER	Bit Error Rate
CALM	Continuous Air-interface,Long and Medium range
CVTCP	Cross-layer Vanet TCP
CWND	Congestion Window
DSRC	Dedicated Short Range Communications
DUPACKs	Duplication Acknowlegments
EBGR	Edge node Based Greedy Routing
ECN	Explicit Congestion Notification
EPDR	Expected Packet Delivery Ratio
ETSI	European Telecommunications Standards Institute
GPSR	Greedy Perimeter Stateless Routing for Wireless Networks
I2V	Infrastructure à Véhicule
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineer
ISO	International Organization for Standardization
I-TCP	Indirect TCP

ITS	Intelligent Transportation System
MAC	Medium Access Control
MANETs	Mobile Ad hoc Networks
MCTP	Mobile Control Transport Protocol
MTCP	Mobile TCP
NS	Network Simulator
OBU	On Board Unit
RFN	Route Failure Notification
RRN	Route Re-establishment Notification
RSU	Road Side Unit
RTO	Retransmission Timeout
RTT	Round Trip Time
SACK	Selective Acknowledgement
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
STT	Short Term Throughput
TCP	Transmission Control Protocol
TCP Bus	TCP BUffering capability and Sequence information
TCP DOOR	TCP Detection of Out-Of-Order and Response
TCP-ELFN	TCP Explicit Link Failure Notification
TCP-F	TCP Feedback
TCP-Sack	TCP Selective Acknowlegement
TCP-W	TCP Westwood

UDP User Datagram Protocol

V2I Véhicule à Infrastructure

VANet Vehicular Ad hoc Network

VTP Vanet Transport Protocol

WAVE wireless access in vehicular environnement