

ral.aux

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ AMMAR TELIDJI - LAGHOUAT

Faculté des Sciences

Département de Mathématique et Informatique



MÉMOIRE DE MASTER

Domaine : Mathématique et Informatique

Filière : Informatique

Option : Réseaux, systèmes et applications répartis (ReSar)

Thème

La Vidéo Streaming dans les réseaux ad hoc véhiculaires VANET

Présenté par :

DADDI MOUSSA IDER MOHAMED

Soutenu devant le jury composé de :

Mr.Y.GUELLOUMA	Président	U. Amar TELIDJI, Laghouat
Mlle.S.BENKOUIDER	Examineur	U. Amar TELIDJI, Laghouat
Mlle.F.BOUSBAA	Examineur	U. Amar TELIDJI, Laghouat
Mr. N.LAGRAA	Encadreur	U. Amar TELIDJI, Laghouat
Mr. Z.NADIR	Co-Encadreur	U. Amar TELIDJI, Laghouat

Année Universitaire /2013/2014

Je dédie ce mémoire

à

mes parents

qui m'ont toujours soutenu et encouragé

au cours de la réalisation de ce mémoire de fin d'étude.

à

tous les professeurs et enseignants universitaires

qui m'ont permis, par leurs efforts, d'atteindre un tel niveau de formation.

Remerciements

En tout premier lieu, Je remercie tout puissant le bon DIEU de m'avoir doté de la santé, la volonté pour venir à bout de ce travail.

J'adresse mon profond gratitude a mon encadreur Monsieur LAGRAA Nasreddine pour ca patience, sa disponibilité, ses expertises, et surtout ses judicieux conseils. Je tien à remercier sincèrement Monsieur ZINELAABIDINE Nadir, qu'il m'a facilité la tache à accomplir mon travail.

Mes remerciements s'adressent également tous les enseignants qui m'ont encadré et conseillé tout le long de mes études.

Je n'oublié pas mes parents, pour leur soutien. Et tous mes amis, et tous ceux qui ont participé de près ou de loin à la réalisation de ce travail. Je tien à remercier aussi les membres du jury d'avoir accepté de juger mon travail, et tout l'encadrement du Département de mathématique et d'informatique.

RÉSUMÉ

Ce mémoire fait l'objet d'une étude de la vidéo streaming dans les réseaux VANeTs (Vehicular Ad hoc NeTworks).

Nous avons commencer par étudier la vidéo streaming dans les différents types des réseaux, ainsi que les solutions proposées pour résoudre les problèmes de la transmission des vidéos.

Une classification des approches proposés pour la vidéo streaming dans VANET, et leurs principe de fonctionnement, avantages et les inconvénients de chacune, sont aussi aussi présentés dans ce mémoire.

Nous avons aussi étudié la performance de l'approche VAPER proposée de ce même contexte, par la simulation en utilisant un ensemble d'outils comme EVALVID.

Mots clés : La Vidéo streaming, VANeT, VAPER, NS2

ABSTRACT

In this thesis, we study the video streaming in VANET (vehicular ad hoc network). First we start with a general description of VANET network, its architecture, its characteristics and application areas

Then in the second chapter we study the streaming video in different types of networks : wired, MANET and tips to solve the problems found

Chapter three is a summary of the proposed video streaming in VANET approaches, their principle of operation and advantages and disadvantages of each one with a classification of these approaches.

The last chapter is an implementation of the protocol VAPER and studies of three criteria of QoS : delay, packet loss and the average rate

Keys words : Vidéo streaming, VANeT, VAPER, NS2

sommaire

1	<i>Introduction au vanet</i>	3
1.1	Introduction	3
1.2	Présentation du réseau VANET :	3
1.3	Caractéristiques des réseaux véhiculaires :	4
1.4	Architecture :	5
1.4.1	La communication Véhicule à Véhicule V2V :	5
1.4.2	La communication Véhicule à Infrastructure V2I :	5
1.4.3	La communication hybride :	6
1.5	Les applications du VANET :	6
1.6	Les travaux de standardisation dans VANET :	7
1.6.1	WAVE(Wireless Access in Vehicular Environments) :	7
1.6.2	L'ISO et le standard Calm :	8
1.6.3	ETSI : TC ITS	9
1.7	Les principaux défis de recherches dans les réseaux véhiculaires :	10
1.7.1	La sécurité	11
1.7.2	L'accès au canal	11
1.7.3	Localisation des véhicules	11
1.7.4	Mobilité dans la simulation des réseaux	11
1.7.5	Routage	12
1.7.6	le vidéo streaming	12
1.8	Conclusion :	12
2	<i>Video streaming dans les reseaux filaires et manet</i>	13
2.1	Introduction	13
2.2	Définition du vidéo streaming :	13

2.3	Les types de vidéo streaming :	14
2.4	La vidéo streaming dans un réseau filaire :	14
2.5	Les protocoles du vidéo streaming :	16
2.5.1	RTP :	16
2.5.2	Les données RTP :	16
2.5.3	Le RTCP and SDP :	17
2.6	La vidéo streaming dans un réseau sans fil et le MANET :	18
2.7	Les techniques de vidéo streaming dans MANET :	21
2.7.1	Les techniques au niveau de la couche MAC :	21
2.7.2	Les techniques au niveau des couches réseau :	22
2.8	Conclusion :	24
3	<i>la vidéo streaming dans VANeT</i>	25
3.1	Introduction	25
3.2	Classification des approches proposés pour VANeT	26
3.3	Le streaming à base d'un codage de réseau :	27
3.3.1	Algorithme de NCDD :	27
3.3.2	Principe de protocole Codeplay :	28
3.3.3	Algorithme de fonctionnement du Codeplay :	29
3.3.4	Procédure de LRR :	30
3.4	Le streaming basé sur le renvoi saut à saut :	31
3.4.1	Principe de fonctionnement de V3 :	31
3.4.2	Algorithme de V3 :	32
3.5	Le streaming basé sur cluster :	34
3.5.1	principe de fonctionnement de VAPER :	34
3.5.2	Algorithme de VAPER :	34
3.5.3	Principe de fonctionnement de ZIPER :	37
3.5.4	Le fonctionnement de TAPR :	38
3.6	(OMV) Overview of the proposed overlay multicast in VANET :	38
3.7	Conclusion	39
4	<i>Simulation et Evaluation des résultats</i>	40
4.1	Introduction	40

4.2	Outil de simulation ns2 :	40
4.3	Génération de mobilité avec IMPORTANT :	41
4.3.1	Modèle de Mobilité Freeway(FW) :	41
4.3.2	Définition d'une carte avec IMPORTANT :	42
4.4	Vidéo simulator Evalvid :	42
4.4.1	Les exigences de base de l'outil :	43
4.4.2	La vidéo choisie pour la simulation :	43
4.4.3	Les étapes de préparations de la vidéo pour la simulation :	43
4.5	L'évaluation du protocole VAPER :	45
4.5.1	Le fichier myUdp.h :	45
4.5.2	Le fichier myUdp.cc :	46
4.5.3	Exporter la classe C++ en OTcl :	46
4.5.4	Exporter les variables C++ en OTcl :	47
4.6	Evaluation des performances :	48
4.7	Conclusion	50
	Conclusion et Perspectives	51
	Bibliographie	52

Table des figures

1.1	les composants d'un véhicule intelligent [1]	4
1.2	les différents types d'architecture dans VANET	6
1.3	pille protocolaire de WAVE(Wireless Access in Vehicular Environments)	8
1.4	le modèle standard Calm	9
1.5	le modèle ETSI	10
2.1	Les types de streaming vidéo	15
2.2	Format de l'entête RTP	17
2.3	Exemple d'un réseau sans fil avec infrastructure	19
2.4	Exemple d'un réseau ad hoc MANET	20
3.1	classification des approches de vidéo streaming dans VANET	26
3.2	Un exemple de structure multicast dans un VANET [2]	26
3.3	Exemple de code play[3].	28
3.4	le calcul de LRR	30
3.5	exemple de procédure du routage de Code Play [3]	31
3.6	exemple de V3 [4]	32
3.7	les chemines des deux types de flux [4]	33
3.8	exemple de l'application DVAC	37
3.9	exemple d'application ZIPER	37
3.10	de fonctionnement de TAPR[5]	38
3.11	exemple de multicast overlay dans un VANET urbain	39
4.1	exemple de fichier qui définit la structure de la carte	42
4.2	Caractéristiques de la vidéo news	43
4.3	exemple de fichier sender trace	44

4.4	L'entête des messages de contrôle dans VAPER	45
4.5	déclaration de classe taitimer	46
4.6	déclaration de CHtimer	46
4.7	Format de l'objet de liaison	47
4.8	déclaration de constructeur	47
4.9	évaluation de délai moyen en fonction de nombre de véhicule	48
4.10	évaluation de moyen des paquets perdus en fonction de nombre des véhicules	49
4.11	évaluation de taux en fonction de nombre des véhicules	49
4.12	évaluation de taux en fonction de nombre des véhicules	50

Introduction Générale

Le développement technologique qu'a vu le monde d'aujourd'hui a touché presque tous les domaines, spécialement le de la communication qui connaît une évolution considérable depuis l'apparition des technologies sans fil.

Pour pouvoir profité de ces technologies dans le secteur des transports, les chercheurs ont pensé à permettre aux véhicules d'établir des liens entre eux, avec ou sans infrastructures installées aux bords des routes, ce qui a permis l'apparition d'un nouveau réseau appelés communément VANET (Vehicular Ad-Hoc NETwork). Ce genre de réseaux ont venu spécialement pour améliorer la sécurité des passagers et des véhicules circulants dans route. Néanmoins, vu que l'homme de nos jours et de plus en plus accro à tout ce qui est relié au confort, le développement de protocoles assurant cette tâche est devenu très important. Plusieurs applications ont été développées pour assurer ce but, par exemple, la détermination du meilleur chemin vers une destination assurant descritères prédéfinis (distance, embouteillage... etc), la publicité sur des produits ou des évènements, les jeux, le chat, la vidéo conférence, ... etc. Le défi essentiel dans un tel réseau très dynamique réside dans les applications nécessitant le vidéo-streaming où un taux de délivrance de paquets doit être assuré dans un délai réduit. Pour cela plusieurs protocoles ont été développés comme CodePlay [6] et V3 [7] où chacun vise à optimiser un facteur spécifique comme la réduction du délai, la minimisation de perte et de retransmission[8] tout en gardant une bonne qualité de vidéo.

Notre objectif dans ce mémoire est en premier temps de comprendre les différentes approches proposé pour la vidéo-streaming dans le contexte des réseaux véhiculaire. Puis, d'évaluer par simulation le protocole VAPER, avec le simulateur de réseaux NS2.

Le reste de ce mémoire est organisé comme suit : Dans le premier chapitre, nous avons présentées réseaux VANETs, leurs caractéristiques, et quelques ses domaines d'applications. Le deuxième chapitre est consacré entièrement à la notion de vidéo-streaming et son

fonctionnement dans les différents types des réseaux avec les difficultés qu'on peut rencontrer. Le troisième chapitre est dédié à l'étude de différentes approches proposées dans les réseaux VANETs. Dans le quatrième chapitre, on a choisi de présenter les résultats de simulation de l'approche choisie.

Chapitre 1

Introduction au vanet

1.1 Introduction

Depuis plus d'une décennie, les communications inter-véhicules sont intégrées dans les systèmes de transport intelligent et elles constituent depuis, un composant fondamental. Ce composant est utilisé pour améliorer la sécurité et l'efficacité sur les routes, diminuer les accidents et fournir un environnement confortable aux conducteurs et aux passagers[9]. Malheureusement, un grand nombre de problèmes doit être résolu pour permettre une intégration efficace et optimale de ces technologies. Chose qui a poussé l'industrie automobile, les opérateurs de réseaux, les organisations et aussi les particuliers à s'investir dans ce domaine.

Dans ce chapitre, nous allons présenter les réseaux VANET leurs principales caractéristiques, et aussi ses différents types d'architecture. Ensuite, nous exposerons brièvement quelques domaines d'application et les différents travaux de standardisation lancés dans le monde.

1.2 Présentation du réseau VANET :

Un réseau VANET (Vehicular Ad-hocNetWork), est un cas spécifique du réseau MANET où les nœuds mobiles sont des véhicules équipés de microprocesseurs, de cartes réseau et de capteurs (voir figure1.1). Comme tous les nœuds dans les réseaux ad hoc, les véhicules peuvent communiquer sans fil entre eux soit directement soit en passant par des stations de base placées tout au long des routes (pour échanger les informations sur

le trafic, demander des services particuliers ou accéder à Internet...).

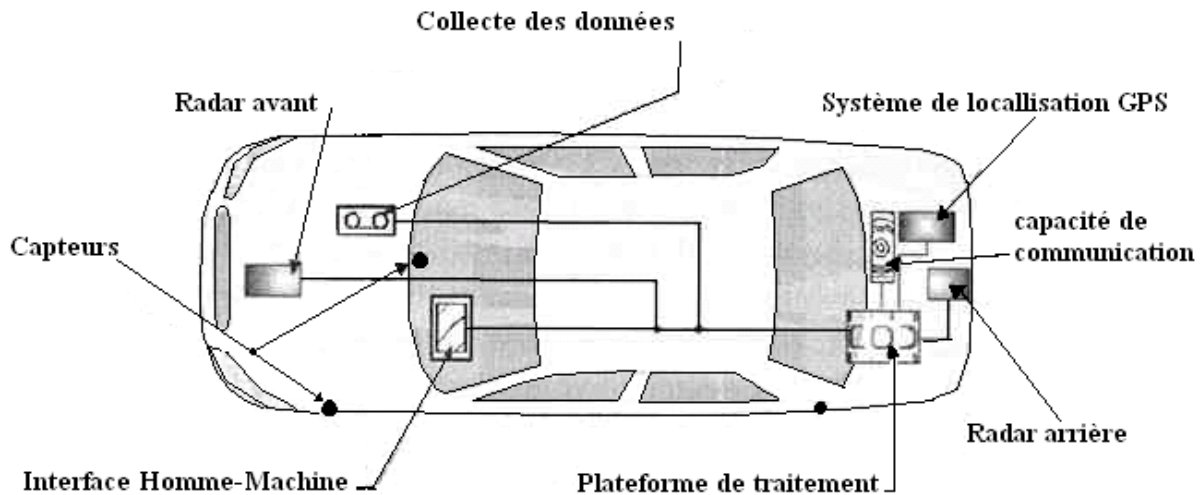


FIGURE 1.1 – les composants d'un véhicule intelligent [1]

1.3 Caractéristiques des réseaux véhiculaires :

Les VANets ont des caractéristiques particulières qui les distinguent des autres types de réseaux sans fil :

- **Energie de communication et capacité de traitement** : Les véhicules dans les réseaux VANets n'ont pas de limite en terme d'énergie et disposent d'une grande capacité de traitement et de stockage [10]. Et peuvent avoir plusieurs interfaces de communication (WIFI, Bluetooth, et autres...).

Mobilité prédictible : Les véhicules ont tendance à avoir des mouvements prévisibles qui sont habituellement limités aux routes. Les informations routières sont souvent disponibles à partir des systèmes de positionnement comme le GPS et la cartographie numérique tel que le MAP.

Scalabilité : les réseaux de véhicules peuvent s'étendre sur la totalité du réseau routier comprenant de nombreux participants.

Forte mobilité : Les environnements dans lesquels les réseaux véhiculaires opèrent sont extrêmement dynamiques. En effet, la densité des véhicules peut être très élevée spécialement durant les heures de pointes, et très faible, pendant les nuits.

Partitionnement : Les réseaux de véhicules sont fréquemment partitionnés. La nature dynamique du trafic peut entraîner des déconnexions importantes entre les véhicules dans

des scénarios moins denses et par conséquent en plusieurs groupes isolés.

1.4 Architecture :

Les services proposés dans les réseaux VANET permettent de distinguer plusieurs types de communication : les communications Véhicules à Véhicules V2V et les communications de Véhicules à Infrastructure V2I. La combinaison de ces deux types de communications permet d'obtenir une communication hybride(voir figure1.2).

1.4.1 La communication Véhicule à Véhicule V2V :

Dans ce type de communication, un réseau de véhicule est vu comme un cas particulier du réseau MANET où les contraintes d'énergie, de capacité mémoire sont relaxées. Cette architecture peut être utilisée dans le scénario de diffusion d'alertes (freinage d'urgence, collision, ralentissement. . .). Un véhicule est en même temps un émetteur, un récepteur, et aussi un relai. Il peut communiquer directement avec les autres véhicules n'importe où, ce qui engendre une communication moins coûteuse et plus flexible. Par contre cette approche souffre de certains inconvénients dont nous citons :

- Les délais de communication sont élevés, étant donné que la communication se fait en utilisant le multi sauts.
- Les déconnexions sont fréquentes à cause de la forte mobilité.
- La sécurité réseau est très limitée.

1.4.2 La communication Véhicule à Infrastructure V2I :

Cette architecture se base sur le passage des paquets de données d'un véhicule à une station de base, située le long des routes, puis ils sont transférés vers un autre véhicule. Cette approche repose sur le modèle client/serveur où les véhicules sont les clients et les stations installées sont les serveurs. Ces serveurs sont connectés entre eux via une interface filaire ou sans fil.

L'inconvénient majeur de cette approche est que l'installation des stations le long des routes est une tâche coûteuse et prend beaucoup de temps, sans oublier les coûts relatifs à la maintenance des stations.

1.4.3 La communication hybride :

La combinaison des deux types précédents, permet d'obtenir une communication hybride très intéressante. En effet, les portées des infrastructures (stations de bases) étant limitées, l'utilisation des véhicules comme relais permet d'étendre cette distance. Dans un but économique et afin d'éviter la multiplication des stations de bases à chaque coin de rue, l'utilisation des sauts par véhicules intermédiaires prend toute son importance.

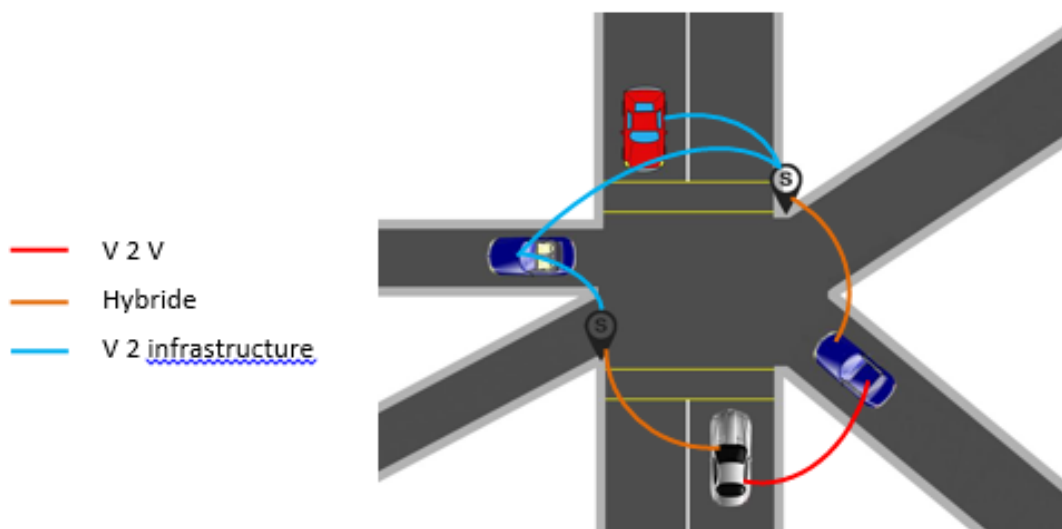


FIGURE 1.2 – les différents types d'architecture dans VANET

1.5 Les applications du VANET :

Le développement des véhicules intelligents a apporté de nouvelles possibilités d'application dans les VANET. Ces applications peuvent être catégorisées en :

- Applications pour la sécurité routière : Dans cette catégorie, on retrouve 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. Nous perdons ainsi les bénéfices de telles applications.

- Applications d'aide à la conduite : pour faciliter la conduite aux gens VANET offre une description générale de l'état des routes autour et la disponibilité de l'espace de stationnement dans les parkings et de coordonner entre automobilistes afin de les guider aux espaces libres. Il s'agit d'améliorer la vision du conducteur en lui proposant une aide à la conduite. 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.
- Applications de confort du conducteur et des passagers : en particulier les services de communication et d'informations des utilisateurs comme l'accès mobile à Internet, la messagerie, le chat inter-véhicules, les jeux en réseaux, vidéo streaming... etc. Ainsi, deux conducteurs par exemple qui voyagent ensemble peuvent s'échanger des messages ou partager des données (vidéo, musique, itinéraire, jeux en réseau).

1.6 Les travaux de standardisation dans VANET :

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).

1.6.1 WAVE (Wireless Access in Vehicular Environments) :

L'IEEE a développé une architecture connue sous le nom de WAVE (Wireless Access in Vehicular Environments), pour offrir l'accès sans fil dans les environnements véhiculaires [11]. (voir figure 1.3). Deux modes de communication sont possibles dans l'architecture WAVE, véhicule-à-véhicule et véhicule-à-infrastructure. Les véhicules communiquent via un dispositif installé à leur bord dit, OBU (On Board Unit).

En 2003 le groupe de travail IEEE a pris la norme sans fil DSRC (Dedicated Short Range Communication) qui était adoptée en 2002 par l'ASTM (American Society for Testing and Materials). Cette norme utilise la bande de fréquences de 5.9 GHz en Europe et aux Etats-Unis et 5.8 GHz au Japon [12].

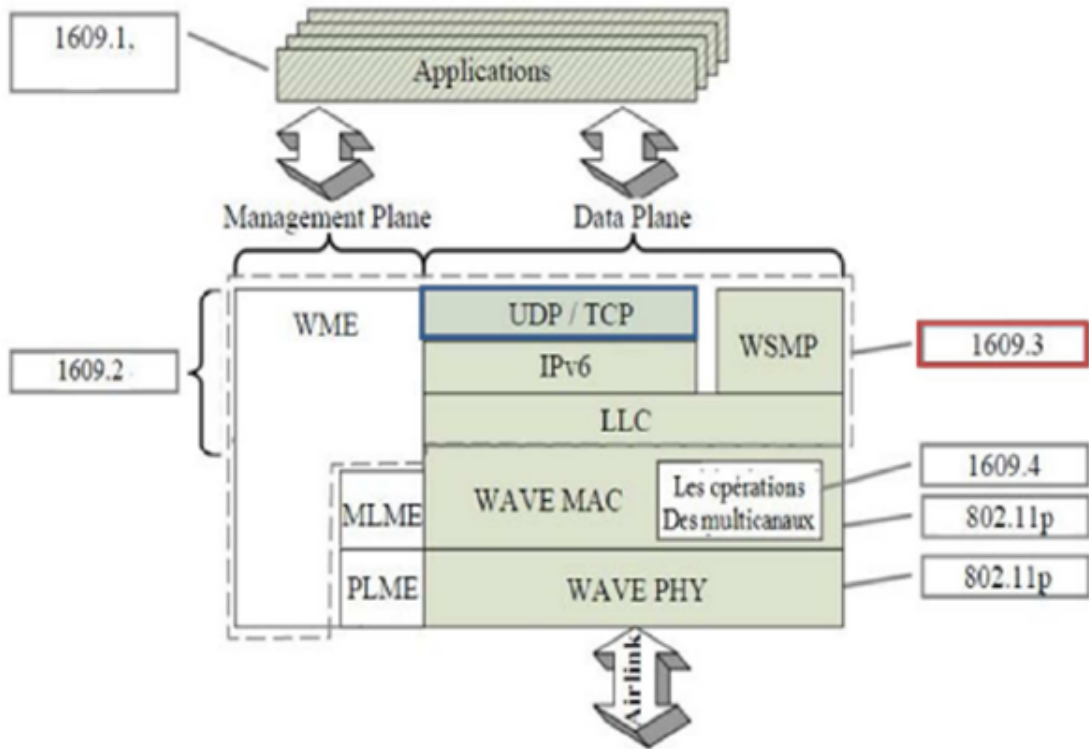


FIGURE 1.3 – pille protocolaire de WAVE(Wireless Access in Vehicular Environments)

1.6.2 L'ISO et le standard 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), dont l'objectif est d'assurer une interface entre plusieurs technologies de communication(voir figure1.4) 2G, 3G, DSRC, ainsi que 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[12].

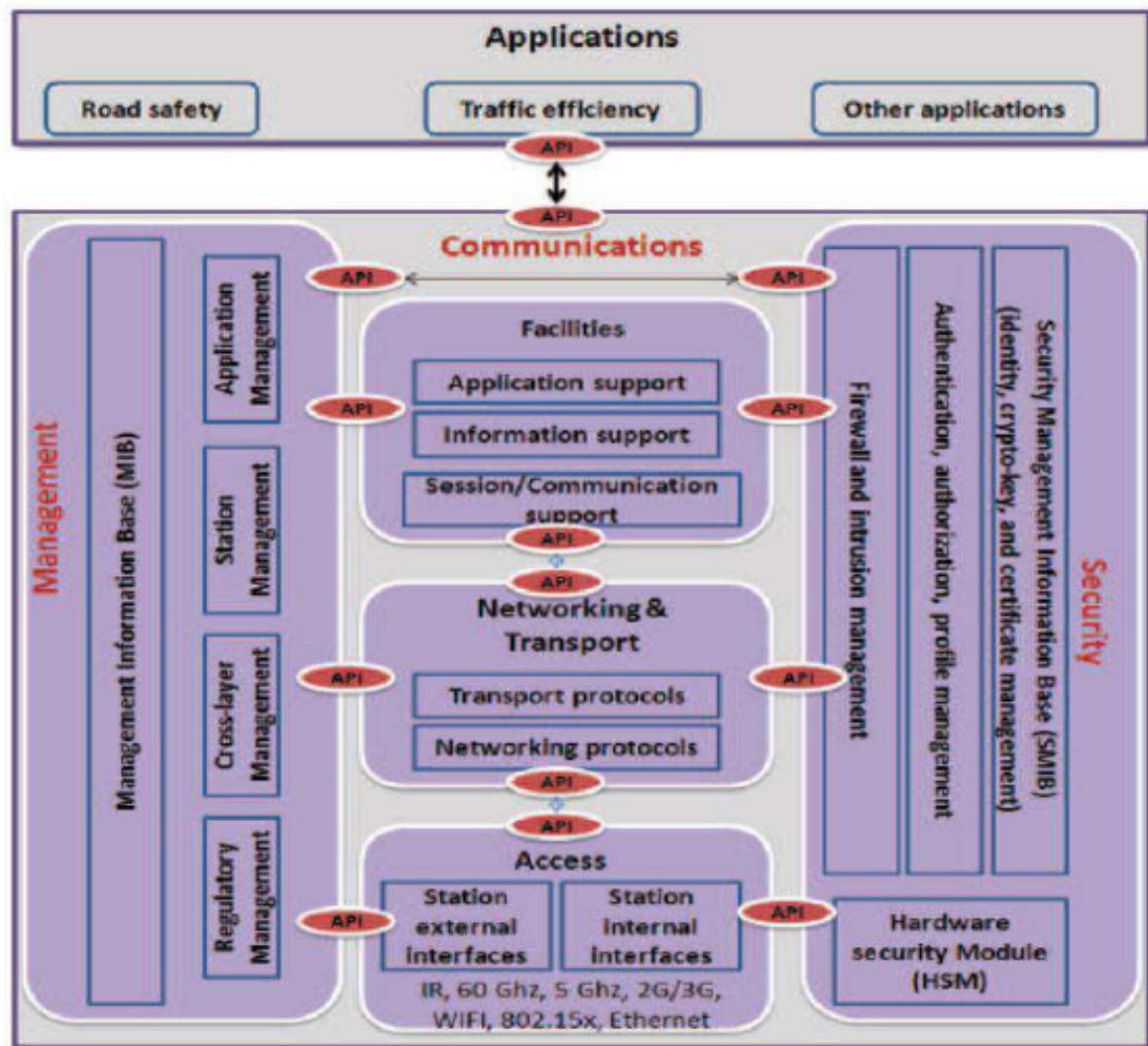


FIGURE 1.4 – le modèle standard Calm

1.6.3 ETSI : TC ITS

Au niveau Européen, ETSI a créé un comité technique TC ITS, afin de développer des standards et spécifications pour les ITS. Le comité a mis en place un plan de route pour les prochaines années, afin de produire un ensemble de standards allant de l'architecture de communication à la spécification de protocoles, et il est organisé en 5 groupes de travail(voir figure1.5) :

- 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.

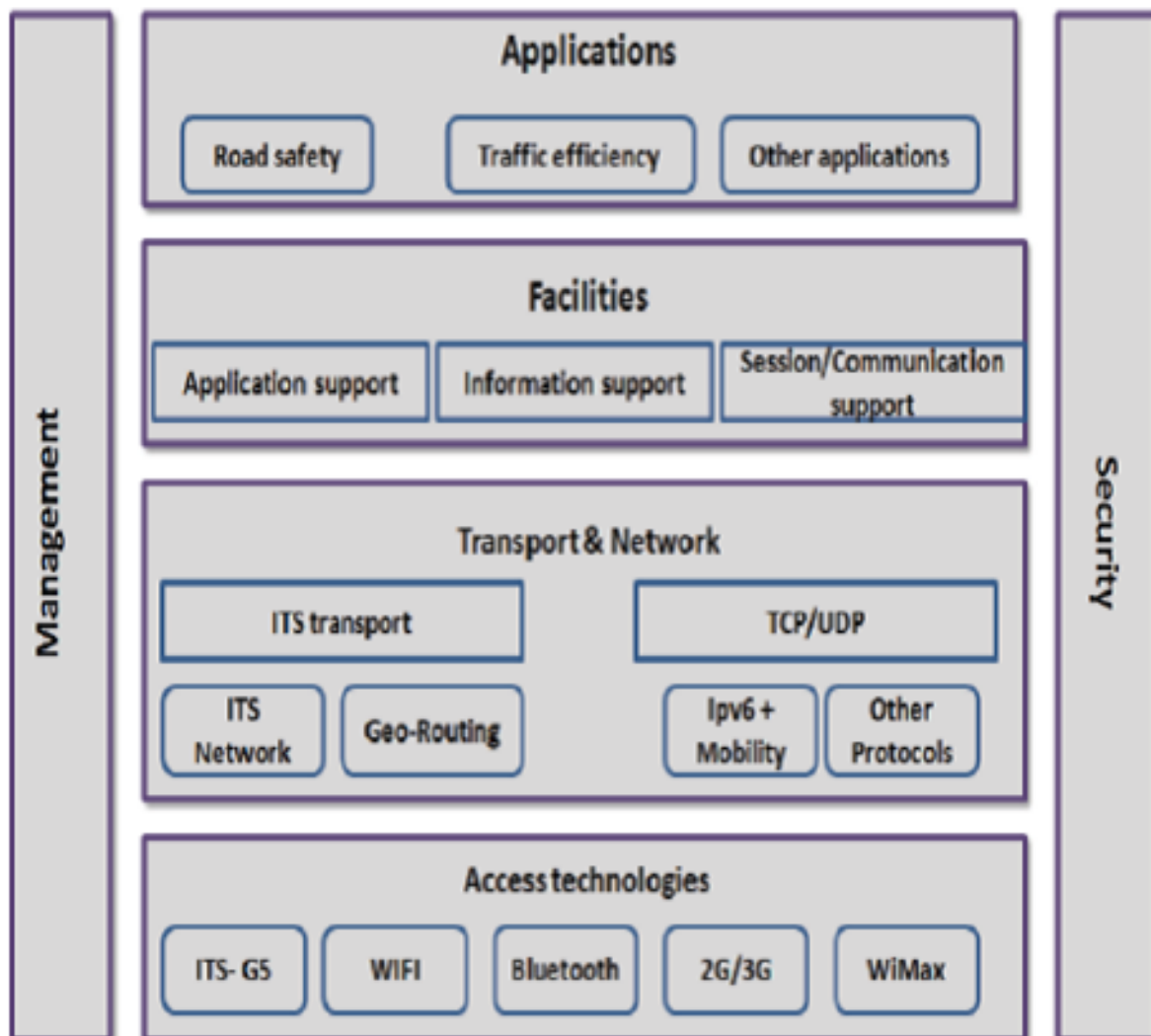


FIGURE 1.5 – le modèle ETSI

1.7 Les principaux défis de recherches dans les réseaux véhiculaires :

La popularité des réseaux véhiculaires a permis de dériver plusieurs axes de recherche dont les plus connus sont :

1.7.1 La sécurité

La sécurité est un défi important ayant un impact majeur sur la mise en œuvre des réseaux véhiculaires ainsi que ses applications. En raison des vulnérabilités des domaines d'utilisation des VANets, l'élaboration des mécanismes de sécurité instituant des relations de confiance entre les entités communicantes (Véhicule ou infrastructure), contrôlant l'accès aux services, garantissant un transfert de données fiable et sécurisé, s'avère d'une importance capitale, car la moindre intrusion d'un véhicule malicieux aurait des conséquences graves sur l'ensemble des véhicules interconnectés[10].

1.7.2 L'accès au canal

Les VANets utilisent les communications radio pour communiquer, par conséquent il est important de concevoir des solutions MAC spécifiques aux réseaux véhiculaires qui permettent d'apporter une certaine qualité de service et gérer les priorités tout en résolvant les problèmes de multi-trajets, d'interférences radio, de l'allocation de ressource distribuée[10].

1.7.3 Localisation des véhicules

La localisation géographique des véhicules est nécessaire, vue que les véhicules ont besoins de cette information en cas d'accident ou en cas d'envoi d'un message. Néanmoins, Le problème est que tous les véhicules ne possèdent pas un GPS intégré d'où la nécessité d'avoir un mécanisme de localisation sans utilisation de GPS[8].

1.7.4 Mobilité dans la simulation des réseaux

La mobilité des véhicules dans une simulation est un facteur important qui peut biaiser les résultats de simulation, donc plus le modèle de mobilité est réaliste, plus les résultats de simulation sont proches de la réalité d'où la précision des modèles de mobilité sur la réussite d'une simulation. C'est pour cette raison on trouve plein d'équipes de recherche qui se penchent sur l'étude de la mobilité dans les VANets[9].

1.7.5 Routage

Le routage est un vaste axe de recherche dans les réseaux véhiculaires et un problème très compliqué à gérer, pour le seule et unique but de permettre aux véhicules de communiquer entre eux, dans ce cas un protocole de routage doit être développé et utilisé dans le cas où les véhicules ne sont pas à une portée de transmission radio directe, le routage est exigé pour établir un chemin (véhicules relais) pour acheminer des données d'un véhicule source à un véhicule destination[10].

1.7.6 le vidéo streaming

Les applications de confort sont devenues un besoin fondamental dans les réseaux. Dans cette catégorie, il y a certaines applications qui sont liées à la satisfaction du conducteur, comme par exemple : le vidéo streaming. Donc, assurer une délivrance avec un débit élevé, un minimum de perte et sans déconnexion est une tâche difficile dans un réseau fortement mobile. Ces critères déclenchent des défis pour le chercheurs travaillant sur le vidéo streaming dans VANET afin d'assurer le maximum de confort aux usagers de la route.

1.8 Conclusion :

VANET soulève un certain nombre de challenges notamment de point de vue protocolaire. Dans ce contexte, nous nous intéressons aux problèmes liés aux applications de vidéo streaming qui doivent respecter certaines normes (débit élevé, pas de déconnexion, QoS ...etc.) tout en s'adaptant au contexte du changement dynamique de la topologie du réseau et aussi à la forte mobilité des véhicules. Dans le chapitre suivant, on s'intéresse au problème du vidéo streaming dans les réseaux filaires et Ad hoc.

Chapitre 2

Video streaming dans les reseaux filaires et manet

2.1 Introduction

Le streaming est le processus de réception et d'affichage de l'information pendant que celle-ci est envoyée par un fournisseur. Les flux transmis et affichés sont du type multimédia (vidéo, audio ou les deux).

Les flux audio et Les flux vidéo sont volumineux, ce qui suppose un volume important de transmission. Cette latence peut ne pas être acceptable pour l'utilisateur[6]. Le flux multimédia peut être envoyé via Internet en utilisant des protocoles connus comme RTP (Real Time Protocol) ou RTSP (Real Time Streaming Protocol). Dans les réseaux filaires classiques, on doit toujours corriger le problème de débit, délai et gigue.

Cependant dans les réseaux mobiles, les défis de mobilité des nœuds et le support de communication non fiables ajoutent aux challenges des réseaux filaires.

Dans ce chapitre, nous allons présenter ce que signifie la vidéo streaming en montrant les caractéristiques de ce service et les différents types. Ensuite, nous exposerons les protocoles spécifiés pour la vidéo streaming dans les réseaux filaires et ad hoc.

2.2 Définition du vidéo streaming :

Le streaming est une méthode de transmission de médias comme un flux continu de données pouvant être traitées par le récepteur avant que le fichier ait été complètement

transmis. La Vidéo streaming est un contenu envoyé sous une forme compressée sur Internet s'affichant en temps réel. Avec la vidéo streaming ou medias streaming, un utilisateur Web n'a pas à attendre pour télécharger un fichier à exploiter[7].

Pour plus de clarté, la vidéo streaming est envoyé comme un flux continu de données qui est traité à l'arrivée. L'utilisateur avec un outil de traitement, qui est un programme spécial décompresse et envoie les données vidéo à l'écran et les données audio aux haut-parleurs. Cet outil peut être soit une partie intégrante d'un navigateur ou un logiciel spécialisé.

La vidéo streaming est généralement envoyé à partir de fichiers vidéo préenregistrées, mais peut être aussi distribué dans le cadre d'une émission en direct (Conférence en ligne). Dans la diffusion en direct, le signal vidéo est converti en un signal numérique comprimé et transmis depuis un site Web particulier[7].

2.3 Les types de vidéo streaming :

De nos jours différents types d'applications du vidéo streaming sont disponibles. On trouve des communications de streaming point-à-point, des communications de broadcast, ou des communications multicast. En plus, les applications de vidéo streaming peuvent être aussi classées en deux types de vidéo streaming (voir figure 2.1) :

1. Le premier est la vidéo streaming en direct, qui est spécifique à la transmission en temps réel d'une vidéo sur Internet. Cette application vidéo est principalement utilisée par exemple dans les diffusions des évènements sportifs en direct.
2. Le second type de vidéo streaming est la vidéo à la demande, ce type est tout à fait contraire au type précédent. Ainsi, une vidéo sélectionnée est jouée à chaque fois que l'utilisateur souhaite regarder la vidéo. Dans ce type de diffusion, un ou plusieurs utilisateurs peuvent voir la même vidéo. Cette approche est principalement observée dans la vidéo streaming des sites comme youtube[13].

2.4 La vidéo streaming dans un réseau filaire :

Une des avantages influant sur le streaming dans un réseau filaire est le débit de données élevé et stable. Ce qui favorise fortement les transferts de fichiers d'un ordinateur vers

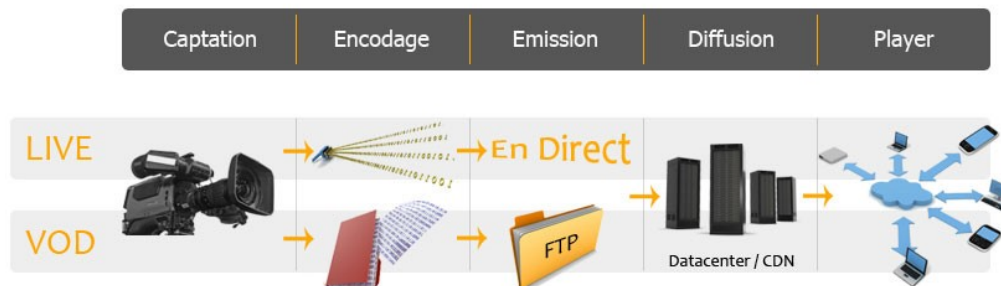


FIGURE 2.1 – Les types de streaming vidéo

un autre; garantissant un vidéo streaming de bonne performance. Le fichier multimédia doit passer, pour son transfert, par plusieurs étapes afin que toutes les informations soient livrées.

1. En vidéo streaming, on tire le signal audio brut et vidéo, puis on les capture sous format de fichiers informatiques.
2. Coder la vidéo capturée dans un format spécifique tel que Windows Media Streaming, QuickTime, Real Vidéo RealNetworks ... etc.

L'encodage est un élément crucial dans la préparation du streaming; c'est là que le débit approprié peut être réglé en gardant à l'esprit que le public a le matériel nécessaire hard et logiciels. Et plus important encore est la vitesse de connexion pour le soutien du streaming.

3. Diffuser le fichier encodé dans le réseau par un serveur de diffusion. Contrairement à un serveur Web, le serveur de streaming contrôle la diffusion de flux en temps réel, et la gestion de la charge d'une manière efficace augmentant ainsi le rendement.

Beaucoup de protocoles, peuvent être utilisés pour la livraison de contenu multimédia comme les RTP, RTSP Les protocoles de transport envoient des flux de bits compressés et des paquets audio / vidéo sur un réseau LAN ou à travers l'Internet. Les paquets délivrés avec succès au récepteur passent d'abord à travers les couches de transport et sont ensuite traitées par la couche d'application, avant d'être décodé dans le décodeur vidéo / audio.

2.5 Les protocoles du vidéo streaming :

Dans le monde du streaming il existe plusieurs protocoles qui ont été développés pour faciliter la diffusion des contenus multimédia en temps réel, parmi eux voici brièvement la description de quelques-uns :

2.5.1 RTP :

Real-Time Transport Protocol, (RTP) qui fournit de bout en bout la fonction de transport dans un réseau pour les applications de transmission de données en temps réel qui sont intolérants au retard, comme l'audio, la vidéo ou les données de simulation. Le RTP ne fournit pas de réservation de ressources et ne garantit pas la qualité de service pour les services en temps réel. Mais, il n'interfère pas dans la transmission. Il permet de :

- Reconstituer la base de temps des différents flux multimédia (audio, vidéo...).
- Détecter les pertes de paquets.
- Identifier le contenu des paquets pour leur transmission sécurisée.

Par contre il ne permet pas de :

- Réserver des ressources dans le réseau
- Apporter une fiabilité dans le réseau.
- Garantir le délai de livraison.

Le RTP peut s'appuyer sur différents protocoles. Dans l'architecture TCP/IP, il s'appuie sur UDP. Il fait partie intégrante de l'application contrairement à d'autres protocoles de transport comme TCP.

Dans une session multimédia, chaque média est transporté dans des sessions RTP distinctes. Cela permet de s'adapter à la bande passante des destinataires où certains peuvent ne recevoir que l'audio. C'est grâce à l'identificateur de la source et à l'horodatage des échantillons que la synchronisation peut être assurée[14].

2.5.2 Les données RTP :

Le RTP transporte les signaux audio ou vidéo en streaming encodés à l'aide de paquets RTP (voir figure 2.2) contenant un header RTP (en-tête) suivi de ces signaux audio ou vidéo. Un paquet RTP est soumis à la couche UDP qui y rajoute un en-tête UDP. L'ensemble est soumis à la couche IP qui y rajoute aussi un en-tête IP. Le datagramme IP est

6. Transmettre les informations de contrôle sur la session (optionnel).

exemple : identifier un participant sur les écrans des participants.

Le SDP est envisagé pour définir les sessions multimédia pour la session de déclaration, session d'invitation etc. Il offre une mise en page pour définir les informations de session pour les membres du réseau. Ces informations comprennent le nom de la session, le type et le format des médias[6].

2.6 La vidéo streaming dans un réseau sans fil et le MANET :

L'utilisation de vidéo streaming dans un réseau sans fil d'Internet devient de plus en plus courante de nos jours (voir figure 2.3). Cette combinaison ramène l'exigence des applications multimédias en temps réel intolérantes aux ruptures. Surtout dans une nature spécifiée de diversité d'obstacles qui coupent les liaisons radio temporellement. Un grand effort est nécessaire pour s'adapter à ce monde, tels que le délai, la gigue, la perte pour que les exigences des applications multimédias puissent être satisfaites par l'instabilité et le manque de fiabilité des liens radio[15].

Le streaming vidéo dans le MANET (voir figure 2.4) est un challenge principalement touché par ces facteurs tels que la mobilité des nœuds, les interférences, les collisions, et beaucoup plus. Le changement dynamique de la topologie provoque la rupture de la connectivité périodiquement ; ce qui résulte en une grande perte de paquets.

La vidéo streaming en temps réel nécessite des techniques spéciales permettant de minimiser les pertes de paquets dans ces types de réseaux non fiables. En plus, l'évolution des appareils mobiles et les réseaux sans fil offrent la plate-forme pour les applications de vidéo streaming sur les réseaux mobiles Ad hoc (MANET) qui nous met devant de nombreux défis. Ils sont traités par différentes techniques[16]. parmi lesquelles nous décrivons les suivantes :

– Délai et Buffers :

Parmi les problèmes de transmission dans un réseau MANET relié à Internet, nous trouvons le délai de transmission entre deux nœuds qu'on considère parmi les facteurs assurant un bon QoS. Nous appelons ce phénomène la gigue du réseau ;et afin de réduire son effet sur la qualité, nous utilisons les mécanismes de mémoire tampon

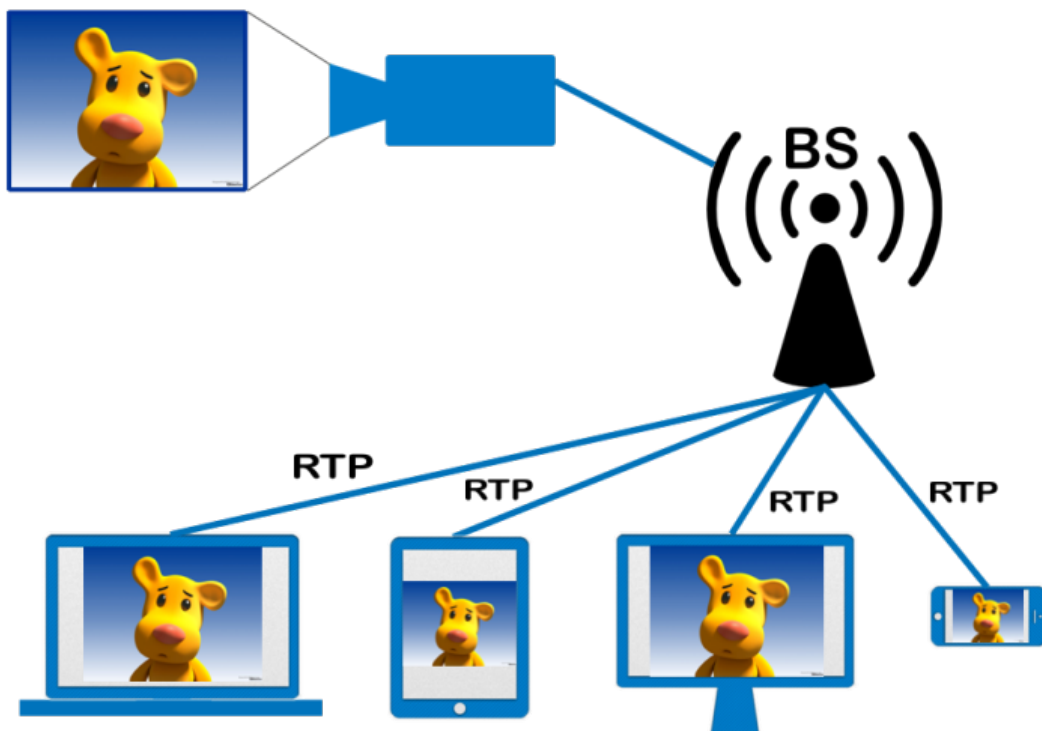


FIGURE 2.3 – Exemple d'un réseau sans fil avec infrastructure

(buffer) côté récepteur. Lorsqu'un paquet vidéo arrive, il n'est pas lu ni traité tout de suite, il reste dans le buffer jusqu'à l'instant de sa présentation.

La taille du buffer est définie en fonction des besoins d'une application de vidéo streaming. Car si celui-ci est trop petit, peut se vider rapidement et peut créer des conflits avec les paquets qui arrivent en retard. Ces sous-remplissages causent un problème dans la visualisation de la vidéo (effet d'images saccadées). A l'inverse, l'utilisation d'un buffer très grand induit des délais de bout-en-bout importants non désirables pour des applications de streaming en direct.

– **SVC Scalable Vidéo Coding et Multiple Description Coding MDC :**

C'est parmi les méthodes de résolution de problèmes de délai où ce type de méthode présente l'avantage de permettre à tous les nœuds du réseau d'avoir leurs ressources utilisées. Même si leurs capacités d'émission sont limitées, à la différence des autres méthodes qui pourraient ne jamais solliciter les ressources d'un nœud jugées trop faibles et solliciter toujours les nœuds à grandes capacités. En effet, même un nœud

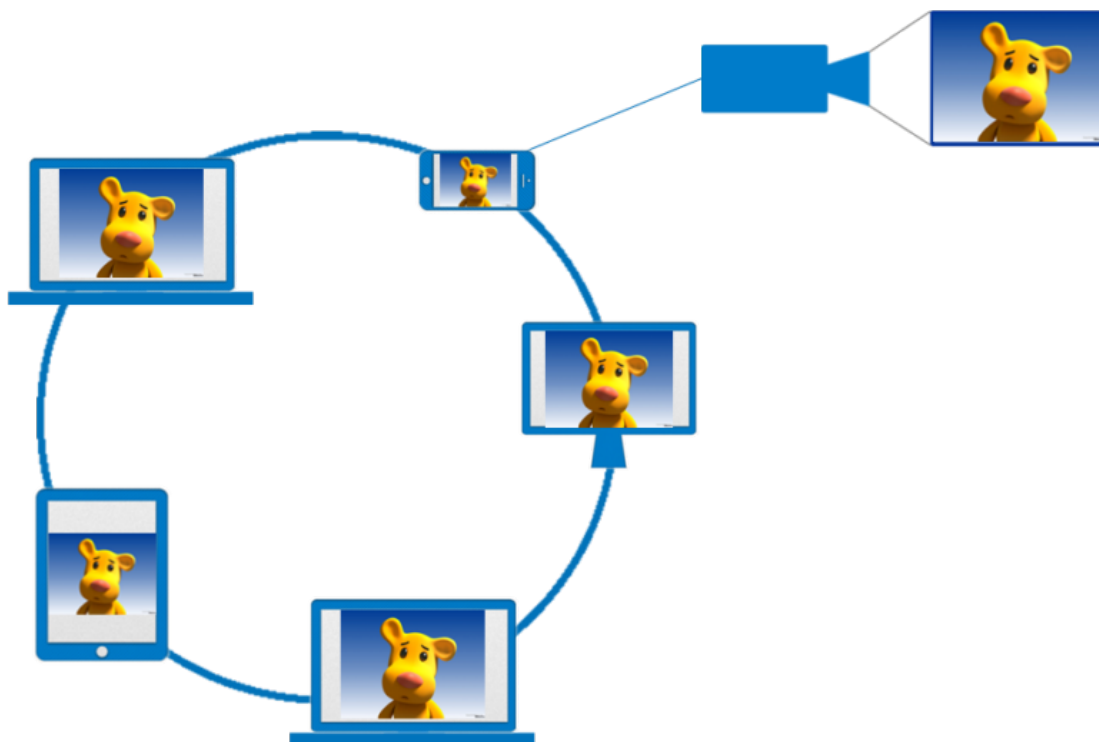


FIGURE 2.4 – Exemple d’un réseau ad hoc MANET

à faible capacité pourrait permettre de diffuser les couches de faibles débits (pour le codage en couche SVC) ou bien un seul niveau de couches (pour le MDC).

SVC (Scalable Vidéo Coding) est une extension du H.264 (MPEG-4 AVC) norme de compression vidéo pour l’encodage vidéo.

Pour assurer une bonne transmission de code vidéo sans dégradation entre divers nœuds mobiles, par exemple entre un ordinateur portable et un appareil mobile ; le SVC traduit les bits à partir d’un flux de données de réseau en une image et qui se traduit à une vidéo (succession des images) dans un flux de bits. Elle décompose le flux de bits vidéo en sous-ensembles des bits en ajoutant des couches de qualité et de résolution des signaux vidéo.

Le SVC peut s’adapter aux connexions réseau en laissant tomber ces sous-ensembles de flux de bits ou de paquets afin de réduire la vitesse de défilement, la résolution ou la consommation de bande passante d’une image, ce qui réduit la gigue. Par exemple : un téléphone mobile ne recevrait que la couche de base ou un flux de bits

en une haute définition. Par contre la console de vidéo conférence recevrait à la fois la couche de base et aussi peu de sous-ensembles ou de couches d'amélioration.

Le MDC (Multiple-Description Coding) est utile en cas de changement de la bande passante et en cas de perte de paquet à cause de la congestion (cas de l'internet).

Le MDC résout de façon efficace le problème de perte de paquets parce que chaque flux binaire peut être décodé de manière indépendante, ce qui rend peu probable d'avoir la même partie de données endommagées dans chaque description.

Le MDC facilite la gestion de la bande passante d'un nœud et le débit, par la transmission d'un nombre approprié de descriptions de l'état de lien de bout en bout. Donc, le MDC peut également exploiter la diversité des chemins[1].

2.7 Les techniques de vidéo streaming dans MANET :

2.7.1 Les techniques au niveau de la couche MAC :

C'est la partie la plus essentielle à travailler pour obtenir des solutions efficaces et optimales. Plusieurs recherches se fondent sur les mises en œuvre des protocoles existant de la famille IEEE 802.11 où des mécanismes sont proposés afin de résoudre le problème de délais, gigue, collision, QoS, et parmi eux :

Le standard 802.11e :

« hybrid coordinator function controlled », channel access (HCCA) et « enhanced distribution channel access » vise à donner à chaque nœud un intervalle de temps pour transmettre ses données. L'IEEE 802.11e spécifique au QoS est une combinaison 802.11a et 802.11g. Il se base sur le fait que la norme 802.11g offre une bande passante plus élevée que la norme 802.11a, Par contre il utilise une bande de fréquence plus élevée, et il a également une transmission plus courte où il offre une couverture évoluée au nœuds mobiles, ainsi que la qualité de service robuste pour la vidéo streaming.

- Au niveau de la couche MAC, on découpe la trame en petits fragments de sous trames pour obtenir un arrangement optimal entre le haut délai des séquences de la trame et de la distorsion à fort potentiel de perdre de grandes quantité d'images vidéo afin de minimiser la perte de donnée lors de l'envoi.
- La limitation de temps de retransmission qui est adapté par le SINR « local signal

to noise and interference ratio » évite de retransmettre des trames qui sont en retard et inutiles dans le but de minimiser la consommation des ressources du réseau.

Les mécanismes de la QoS :

- La différenciation de service : Au niveau de la couche MAC, la définition de plusieurs classes de trafic offrant un accès différencié au canal de transmission est le principe de base de la différenciation de service. Plusieurs paramètres peuvent être utilisés pour construire ces classes de trafic. Parmi ces paramètres : Arbitrary IFS (AIFS), L'opportunité de transmission, Le temps de reprise (Backoff Time BT).
- La garantie de service : Pour mieux partager et contrôler les ressources disponibles c'est mieux d'utiliser le mécanisme de garantie de service. Les deux mécanismes : le contrôle d'admission et la réservation de la bande passante forment la base de la garantie de service. Le contrôle d'admission c'est pour déterminer l'état du réseau et estimer les ressources disponibles. la réservation de la bande passante c'est pour allouer la taille nécessaire de la bande passante pour garantir un meilleur passage de données.

2.7.2 Les techniques au niveau des couches réseau :

Dans les réseaux MANET, la tâche de routage au niveau de couche réseau est d'établir et de maintenir une ou plusieurs routes de bout en bout de la source à la destination ; et qui assure que la vidéo est livrée avec une qualité perceptive satisfaisante. Dans ce qui suit, nous distinguons les techniques de routage où une seule voie est établie et celles dans lesquels le flux de données vidéo sont transmises à travers de multiples chemins. De plus, nous incluons les techniques de streaming multicast par le routage multicast. Parmi ces techniques nous avons :

Routage mono-chemin (Single-path routing) :

Les recherches dans le domaine de routage au réseau MANET engendre trois types de classification des protocoles comme suit :

- Les protocoles proactifs : maintenir au début toutes les informations de routage (si possible) pour tous les autres nœuds. Parmi ces protocoles, nous avons : le OLSR et le DSDV.

- Les protocoles réactifs : obtenir les informations d'autre nœud au fur et à mesure d'une façon dynamique, qui sont le DSR « Dynamic Source Routing », et l'AODV « Ad Hoc On-demand Distance Vector routing ».
- Les protocoles hybrides : c'est ceux qui exploitent la force des deux protocoles réactifs et proactifs à la fois pour la découverte de route la ZRP « Zone Routing Protocol ». La découverte des routes se fait de manière proactive entre les nœuds voisins et de manière réactive entre les nœuds périphériques (ganglions qui sont à l'extérieur du rayon de la zone, mesurée en nombre de sauts).

Ces deux types de protocoles classiques utilisés dans le MANET ont des inconvénients qui ne satisfont pas les demandes de vidéo streaming, c'est pour ça que nous devons étudier les protocoles de routage adaptés au vidéo streaming. En général, ces protocoles sont distingués des protocoles de routage du MANET classiques, par plusieurs mesures utilisées pour déterminer la qualité de chaque route.

Routage multi-chemins (Multipath routing) :

Parmi les techniques les plus importantes pour la vidéo streaming sur le MANET, la mise en place de multiples voies de la source à la destination. En général, un routage multipath peut améliorer la qualité de service en fournissant les éléments suivants :

- Agrégation de bande passante et de retard : misant sur la capacité de plus d'un itinéraire.
- L'équilibrage de charge : équilibrant la charge de trafic entre un plus grand nombre de nœuds et de liens.
- Tolérance aux pannes : ajoutant la redondance, pour réduire l'effet de pannes de réseau sur la qualité de la vidéo transmise, il est important de savoir que les chemins sont disjoints. Dans le cas où le protocole de routage multipath offre de multiples chemins avec une diversité de chemins suffisants, il est moins probable qu'une défaillance de liens affecte divers chemins simultanément. Cela est particulièrement bénéfique pour le streaming en temps réel, où la mémoire tampon de lecture est limitée et le codeur vidéo ne peut plus compter que sur la diversité du temps imparti.

Routage avec QoS (QoS routing) :

Le plus court chemin en termes de distance physique ou nombre de sauts n'est pas toujours le meilleur pour un flux de vidéo en live. La transmission d'un vidéo streaming repose sur les critères assurant la QoS. Les propriétés qui sont utilisées pour déterminer la route la plus appropriée sont décidées selon les paramètres de la QoS (la bande passante, la durée de vie, le retard, la gigue). Des recherches existent pour la QoS dans le MANET, comme exemple : un cadre de QoS de routage dans l'Internet, Richard et Liu[17] proposent un protocole de routage de qualité de service, qui est une extension du DSDV, où la bande passante est considérée comme le critère le plus critique en temps réel. Le but de la solution est d'identifier le plus court chemin avec une bande passante disponible au-dessus de l'exigence minimale. La solution est basée sur l'utilisation de la TDMA (Time Division Multiple Access)[18].

2.8 Conclusion :

le comportement de vidéo streaming dans un réseau filaire est totalement différent par rapport au réseau ad hoc, a cause de ses caractéristiques(mobilité, obstacle. . .) se qui fait que le streaming est difficile a contrôler d'une manière parfaite; qui nous oblige a le réadapter ces caractéristiques pour pouvoir assurer une bonne circulation des flux en temps réel surtout pour les VANET qui ont spécifié d'un forte mobilité. Pour ce la on doit étudier les approches proposés dans ce domaine a fin de minimiser au maximum le nombre des paquets perdus et une transmission dans un délai très réduit.

Chapitre 3

la vidéo streaming dans VANeT

3.1 Introduction

Le Vidéo streaming est l'un des domaines d'application du VANET, varié en plusieurs sous domaines comme vidéo à la demande, informations touristiques, publicité contextuelle et d'autres questions concernant la sécurité comme l'appel vidéo d'urgence.

Le VANET comporte plusieurs avantages pour la vidéo streaming « l'énergie, la mémoire de stockage ... ». Mais le problème qui s'impose c'est la transmission sans fil entre les nœuds tenant compte des grandes vitesses, des problèmes de l'atténuation, de la perte de paquets, et de la variation rapide de la topologie, ce qui influent négativement aux conditions de vidéo streaming « le grand volume de contenu, le délai de service de livraison, consommation de bande passante ».

Dans ce chapitre, on discutera les différents approches proposé pour atteindre un bon gestion de circulation des flux de vidéo streaming dans un réseau VANET au but d'assuré les performances de QoS.

3.2 Classification des approches proposés pour VANeT

après l'étude des différentes approches proposées pour la vidéo streaming dans VANeT, on a proposé cette classification suivante :

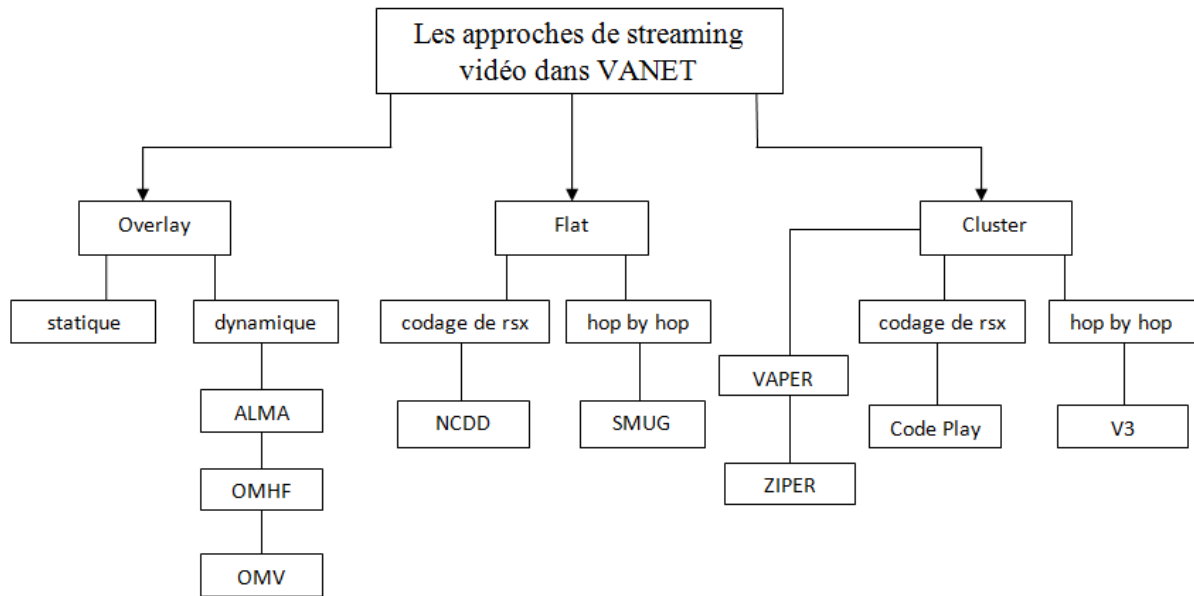


FIGURE 3.1 – classification des approches de vidéo streaming dans VANET

La question qui se pose est : Es-ce que le streaming à la demande peut être supporté par une diffusion en direct ou non? Des mécanismes sont en considération tels que la retransmission et la pré-lecture . Comme le montre la Fig(3.2)

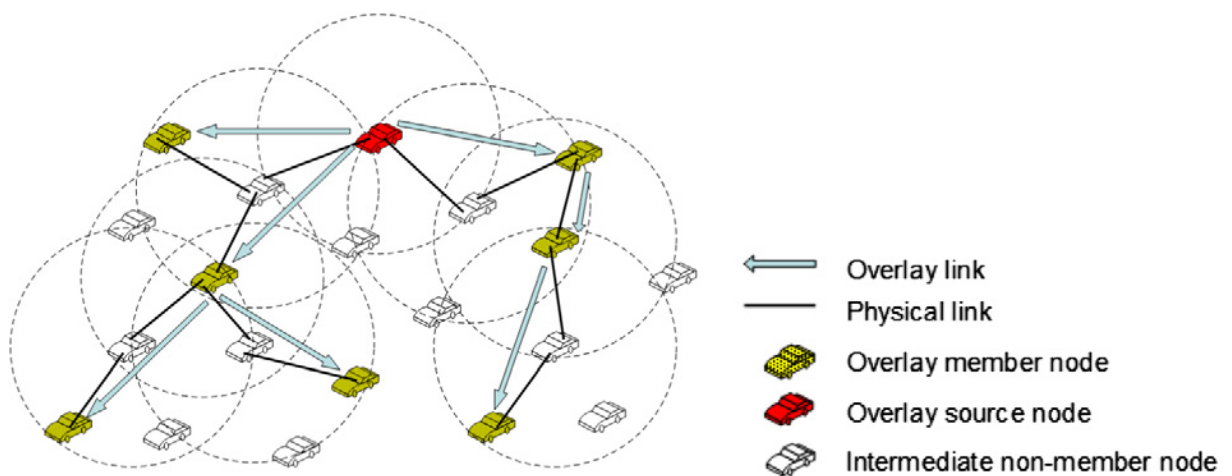


FIGURE 3.2 – Un exemple de structure multicast dans un VANET [2]

Pour qu'un véhicule puisse rejoindre un groupe organisé de façon arborisant où la vidéo streaming existe à la racine de l'arbre. L'avantage principal dans cette structure est qu'un groupe de nœuds a le rôle pour soutenir la construction de recouvrement de multicast et l'entretien. Ce qui rend la restructuration du groupe inutile chaque fois ou un nœud veut le rejoindre. Un avantage est que le recouvrement peut encore bien travailler, même si les nœuds de groupe ne sont pas assez dense. Un autre avantage d'utilisation d'une telle structure, c'est l'acheminement flexible. La livraison de paquets entre les nœuds du groupe est assurée par un protocole de routage sous-jacent qui peut faire face à une meilleure dynamique VANET mieux.

3.3 Le streaming à base d'un codage de réseau :

Le NCDD (Network Coding based Data Dissemination) est une technique qui se base sur la diffusion aléatoire où chaque nœud envoie ses informations aux nœuds voisins à un saut d'une façon périodique. Chaque fois qu'un nœud reçoit une partie (morceau) des données qui n'existe pas dans sa mémoire locale, il la stockera jusqu'à recueillir suffisamment de morceaux pour ensuite les décoder ; mais cette technique n'assure pas que les morceaux envoyés seront bien reçus par les nœuds voisins « perte de paquets ». Donc, on peut tout de même souffrir de collisions à cause de l'envoi simultané des morceaux de données. L'autre inconvénient majeur de cette technique est le temps d'attente pour la réception de suffisamment de parties pour les décoder. Surtout que dans un milieu urbain contenant des obstacles, la communication d'informations est difficile entre les nœuds d'un même groupe.

3.3.1 Algorithme de NCDD :

Au niveau d'émetteur () :

Début :

TQ vrai faire

Broadcast l'information par un seul saut.

Broadcast les données par un seul saut.

Fait

Fin

Au niveau de récepteur() :

```

Si le paquet n'est pas encore reçu alors
Stocker le paquet
Finsi
Si je peux décoder la vidéo alors
Générer et transférer les nouveaux paquets
Finsi

```

3.3.2 Principe de protocole Codeplay :

Le Codeplay résolu le problème de collisions dans le réseau en adoptant un plan d'impulsions locales où seulement certains nœuds sélectionnés sont autorisés à transmettre des paquets vers des nœuds se trouvant dans le même itinéraire.

Dans Codeplay, les routes dans un environnement urbain sont découpées en segments. Chaque segment a un responsable qui s'appelle coordinator ou un relai qui se charge par la retransmission des paquets au segment suivant (voir la Figure 3.3). Chaque nombre de segments ayant leurs portées non fusionnées ont un intervalle de temps pour envoyer ses paquets d'une façon simultanée(voir la Figure 3.5).

La bande passante du Codeplay est découpée en deux canaux, l'un pour les messages de contrôle et le second pour les services ; plus précisément pour le streaming[3]. Néanmoins, cette technique ne corrige toujours pas le problème d'un groupe clairsemé (peu dense)[2].

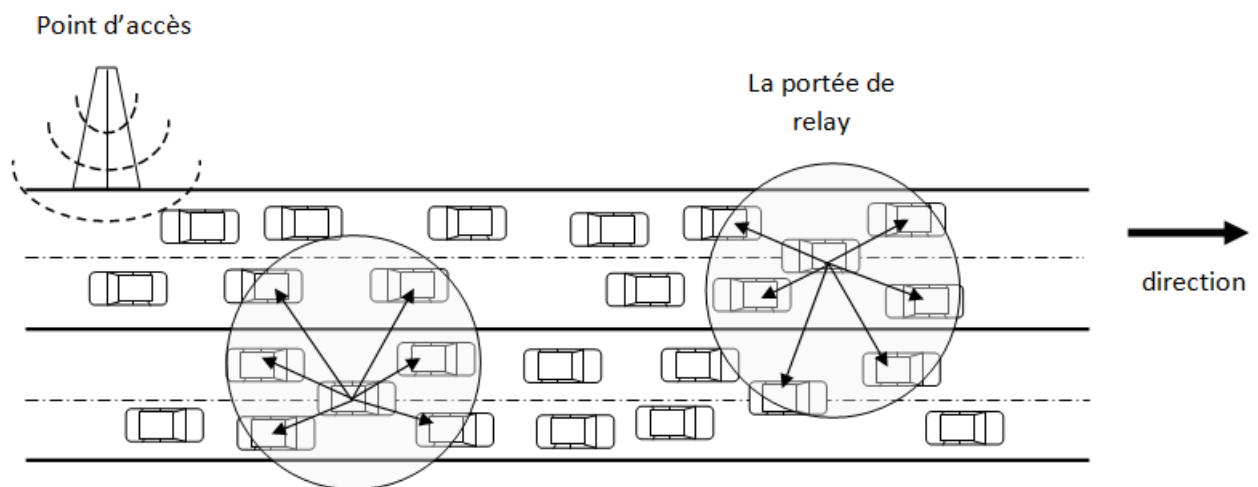


FIGURE 3.3 – Exemple de code play[3].

3.3.3 Algorithme de fonctionnement du Codeplay :

- Tâche du point d'accès :

Début :

-Découper la route en segments.

-Définir les segments concurrents par le LRR (local round-robin).

-Fournir des paquets de streaming pour les véhicules entrant au zone.

Fin

- Tâche de coordinateur :

Début :

Pour chaque slot faire

-Choisir le nœud relai avec le segment voisin suivant les avantages de chaque nœud qui sont transmis par les messages de contrôle.

-Indiquer le nœud relai en envoyant le un message unicast.

-Transmettre les paquets aux autres nœuds du segment. Parmi eux le relai qui lui-même faire les passer au relai voisin.

Si les segments concurrents sont vides alors

$T_i = T_i * 3(\text{nbrs} + 1)$.

-Le coordinateur diffuse un petit signal pour chercher un autre relai dans un intervalle de temps $i \bmod(\text{nbrs}+1)$.

Finsi.

Fait

Fin.

- Tâche des nœuds de segment :

Début :

-Choisir le coordinateur par la voie la plus proche nœud au centre de segment où chaque nœud de segment diffuse sa distance au centre.

Fin.

- Tâche du relai :

Début :

Pour chaque slot faire

-Connecter avec le relai voisin par le LRR

(local round-robin) (voir la Figure 3.4).

-Envoyer les paquets de streaming au relaiy voisin.

fait

Fin

3.3.4 Procédure de LRR :

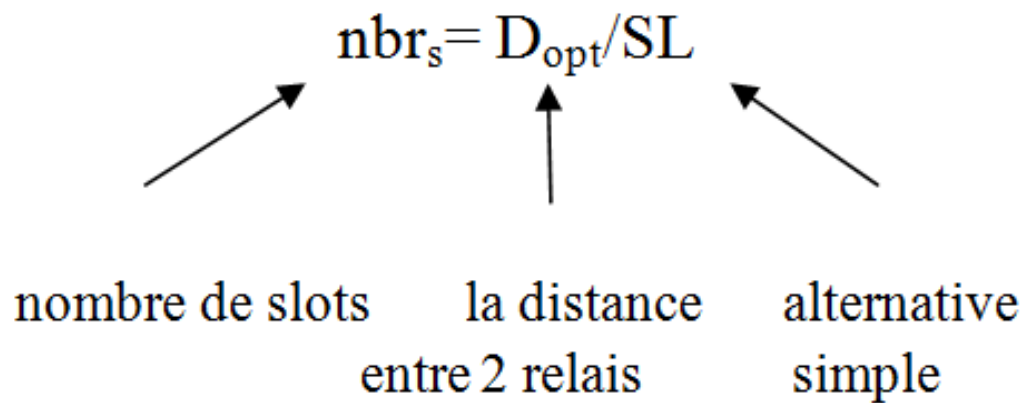


FIGURE 3.4 – le calcul de LRR

nbr_s c'est le nombre de slot entre deux relais concurrents dans un temps T_i $T_i = i \bmod (\text{Nbr}_s + 1)$

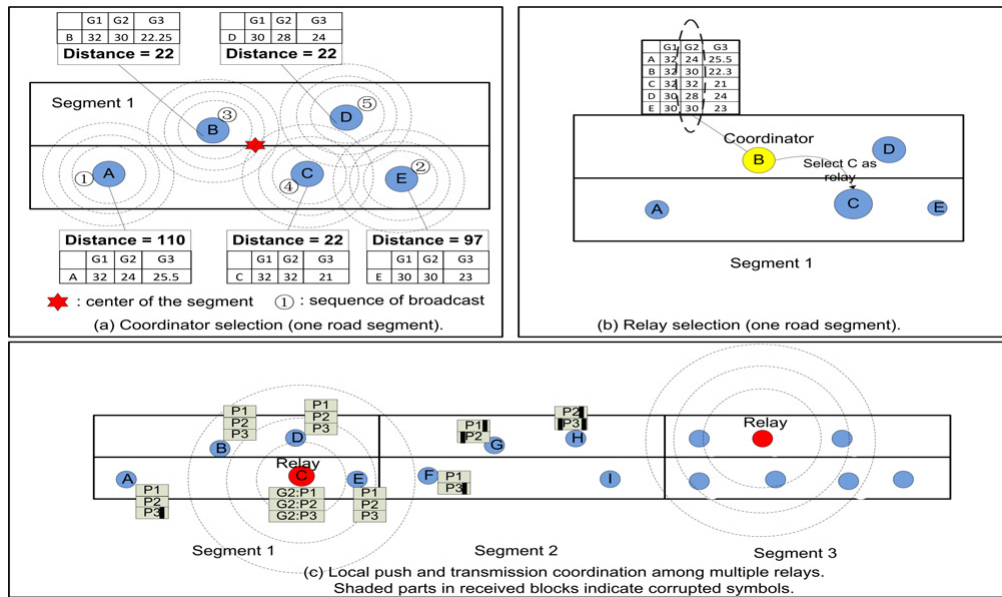


FIGURE 3.5 – exemple de procédure du routage de Code Play [3]

3.4 Le streaming basé sur le renvoi saut à saut :

Dans le SMUG (Streaming Media Urban Grid) : des nœuds sont sélectionnés pour former « SMUG nodes ». Donc quand un flux est généré par un point d'accès. Ce dernier envoie ses paquets à un nœud SMUG. Chaque nœud peut dynamiquement être sélectionné en tant que transitaire, et ses transmissions sont prévues en utilisant un schéma TDMA. Mais le problème se pose quand il n'y a pas assez de nœuds SMUG, car en résultat, on aura une perte de paquets élevée. L'utilisation du TDMA pénalise aussi l'utilisation du SMUG, car le problème de synchronisation est difficile à résoudre.

3.4.1 Principe de fonctionnement de V3 :

V3 fournit un schéma pour qu'un véhicule dans une zone d'intérêt capture et envoie des séquences de vidéos demandées par d'autres nœuds. Le scénario d'application de V3 repose sur le fait que sur une route la scène peut être capturée par une ou plusieurs sources vidéo (stations et véhicules se trouvant avant). Les véhicules concernés sont appelés récepteurs (voir la Figure 3.6).

Ce protocole se base sur deux types de messages, l'un pour les demandes vidéo et l'autre c'est des vidéos capturées. Un nœud récepteur envoie une demande qui passe d'un véhicule à un autre jusqu'à ce qu'il atteigne un véhicule de la zone de destination (voir la Figure 3.7), ce procédé s'appelle vidéo triggering sub-system. Un véhicule ayant reçu la

demande dans la zone d'intérêt, commence à capturer et à envoyer la vidéo demandée au récepteur, cette procédure s'appelle vidéo transmission sub-system[4].

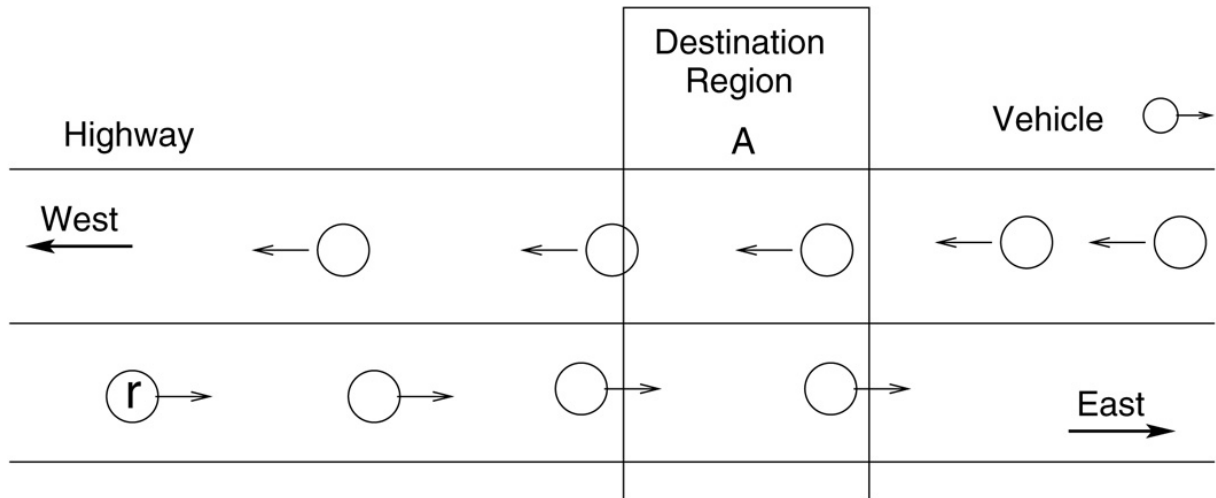


FIGURE 3.6 – exemple de V3 [4]

Ce figure ne convient pas aux communications du groupe, car chaque récepteur établit un chemin d'accès à une source, qui est inefficace, autrement dit, le paquet du protocole de transfert en V3 ne considère que les véhicules dans une route droite, comme une autoroute. Par conséquent, V3 n'est applicable pour le cas d'un espace urbains.

3.4.2 Algorithme de V3 :

- Récepteur :

Début :

Broadcast un message de demande

TQ existe paquets de données faire

Recevoir les paquets de données.

Fait

Fin

Véhicule dans la zone de destination :

Début :

TQ existe demande faire

Recevoir la demande.

Capturer un partie vidéo

Envoyer le a un nœud transitaire

Fait

fin

transitaire :

début :

TQ donnée entrant faire

Si donnée n'existe pas dans la mémoire alors

-Recevoir la donnée.

-chercher un autre transitaire dans sa portée.

Envoyer le la donnée.

Finsi

Fait

TQ demande entrant && temps pas fini faire

Si il n'existe pas dans la zone de passe la demande alors

Recalculer .

Finsi

Inonder la demande aux nœuds qui sont dans la portée.

Fin

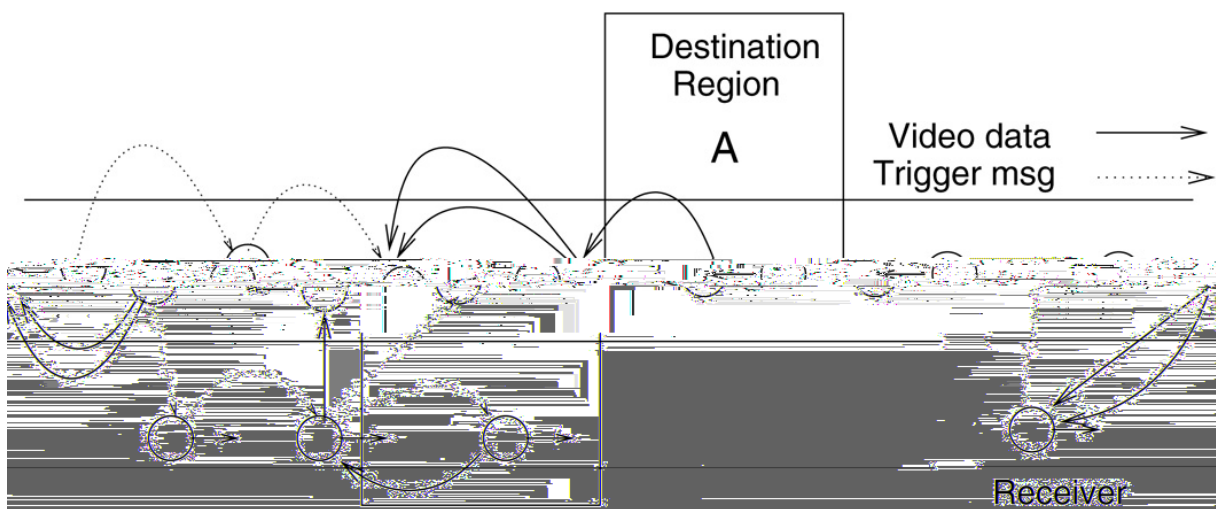


FIGURE 3.7 – les chemines des deux types de flux [4]

3.5 Le streaming basé sur cluster :

Les deux protocoles, VAPER (Véhicules Adaptive Peer-to-Peer Relay Method) et ZIPPER (Zero-Infrastructure P2P System), forment des groupes entre les véhicules où des flux multimédia sont envoyés et relayés entre les clusters. Chaque véhicule envoie périodiquement une balise à ses voisins pour former des clusters, chaque cluster a un leader et un nœud relai avec le cluster voisin.

La différence entre VAPER et ZIPPER se réside dans le fait que VAPER envoie un flux multimédia tandis que ZIPPER fait passer le flux en cours. En VAPER, le leader d'un cluster diffuse un flux multimédia à ses membres, puis le nœud relais du groupe relaye le flux multimédia au leader du cluster suivant.

Dans ZIPPER un flux multimédia est sauvegardé sous forme de blocs, et un véhicule possédant des blocs qui sont demandés par les autres véhicules peut les envoyer dans le réseau.

3.5.1 principe de fonctionnement de VAPER :

VAPER se base sur les deux notions : cluster-head pour le broadcast du vidéo streaming et le cluster-tail pour le faire passer au cluster voisin ; et pour identifier un couple de nœuds, on utilise la procédure de HOOK qui contient six types de messages de contrôle[19] :

1. REQ : utilisé par un membre de cluster pour demander la vidéo streaming.
2. REPLY : utilisé par le propagateur comme une réponse à REQ.
3. REQ HOOK ON : utilisé par le cluster-head ou cluster-tail pour se raccrocher en terme de couple, et ils ont répondu par un REPLY HOOK ON .
4. REQ HOOK OFF : utilisé par le cluster-head ou cluster-tail pour se relâcher en terme de couple, et ils ont répondu par un REPLY HOOK OFF.

3.5.2 Algorithme de VAPER :

Pour exécuter le protocole VAPER pour la vidéo streaming, il faut d'abord exécuter le protocole DVAC « distributed Vehicles-Adaptive Clustering » qui est utilisé pour former le mécanisme de communication entre les différents types des nœuds et pour la maintenance

des clusters (voir la Figure 3.8). Avant d'écrire l'algorithme on définit les messages qui circulent dans le réseau :

1. CH, utilisé par un nœud pour informer ces voisins qu'il va être le cluster-head.
2. CT, utilisé par le cluster-head pour informer ses voisins qui va être le cluster-tail.
3. JOIN, utilisé pour informer ses voisins qu'il va rejoindre un autre cluster.
4. RESIGN, utilisé pour notifier un véhicule pour rejoindre un autre groupe et démissionner de son rôle parce que la taille actuelle du cluster est plus grande que K .

Etape de clustering

Si un nœud est entrain de quitter le cluster alors

Si le nœud est cluster Head alors

Le CT informe les autres nœuds pour choisir un nouveau CT.

Envoie un message Join pour rejoindre un autre Cluster.

Finsi

Si le nœud est cluster-Tail alors

Le CH détecte la sortie d'un CT, et envoie un message CT aux membres pour choisir un nouveau CT.

Envoie un message Join pour rejoindre un autre Cluster.

Finsi

Finsi

Tantque CH reçoit un Join faire

Si le nœud appartient au cluster alors

Si $\text{nbr véhicules} < K$ alors

Envoyer une réponse positive pour rejoindre le cluster

Sinon

Envoyer RESIGN pour l'informer pour rejoindre un autre cluster

Finsi

Finsi

Finsi

Etape du vidéo streaming

Propagator

SI je reçois REQ faire

Si je suis pret alors

Envoyer en réponse un REPLY

Finsi

Finsi

SI je reçois un REQ HOOK ON alors

Si la vidéo est trouvé alors

Envoyer un REPLY HOOK ON

Répéter

envoyer les paquets de la vidéo

Jusqu'à le streaming est terminé ou je reçois REQ HOOK OFF

SI je reçois un REQ HOOK ON alors

Envoyer un REPLY HOOK OFF

Finsi

Sinon

Passer la demande au cluster voisin

Finsi

Fin

Membre

Envoyer un REQ

Si reçoit un REPLY alors

Envoyer un REPLY HOOK ON

Répéter

Recevoir les paquets de la vidéo

Jusqu'à le streaming est terminé ou je décide d'arrêter le streaming

Envoyer REQ HOOK OFF

Finsi

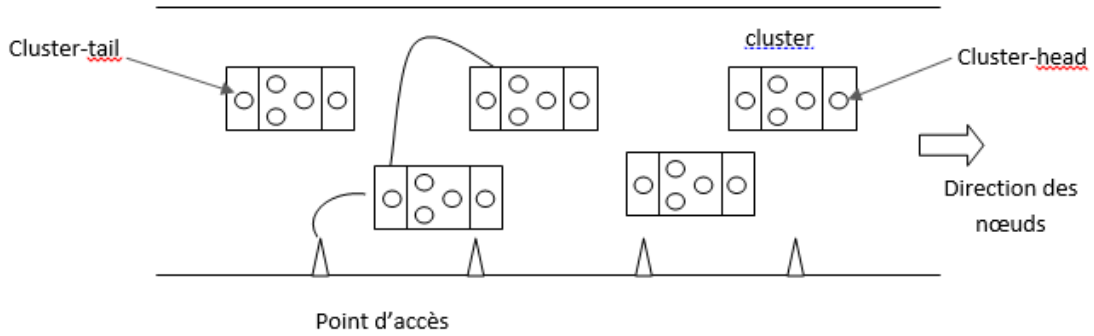


FIGURE 3.8 – exemple de l'application DVAC

3.5.3 Principe de fonctionnement de ZIPER :

L'idée de base se repose sur la recherche des blocs d'un fichier vidéo ou audio dans les véhicules voisins à un seul saut, et s'il ne trouve pas ; il renvoie la recherche des blocs manquants aux autres véhicules en passant par les voisins à un saut (voir figure 3.9). Chaque véhicule contient un bloc, il répond au demandeur et il lance la transmission de streaming en utilisant le protocole TAPR.

La procédure de recherche et de transmission base sur la notion des threads où un nœud peut envoyer la recherche ou répondre à une demande et aussi transmettre des données simultanément[5].

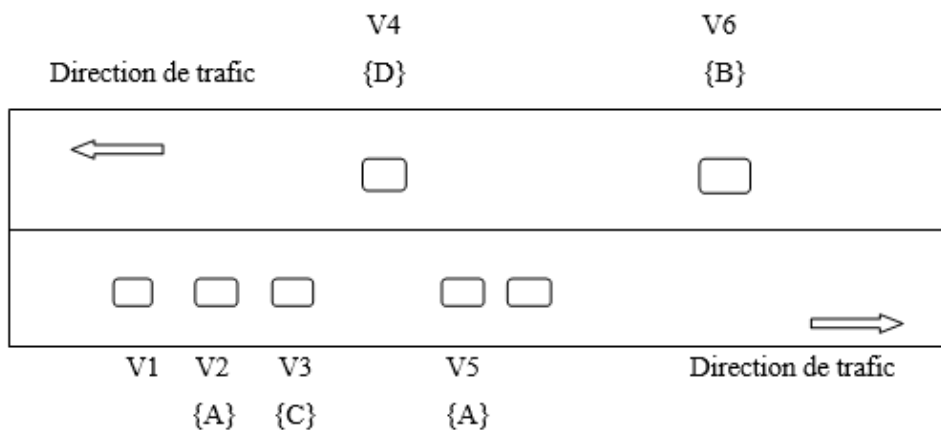


FIGURE 3.9 – exemple d'application ZIPER

3.5.4 Le fonctionnement de TAPR :

TAPR se base sur la notion de cluster où chaque nœud envoie un message de découverte dans une portée de 200 à 300 m chaque 300ms pour détecter ses voisins. TARP utilise les informations de ces messages pour la structuration et la maintenance du cluster .

TAPR résout le problème de déconnectivité(voir la Figure 3.10), en cas où le nœud source envoie un paquet qui ne peut passer au destinataire, il le renvoie au cluster-head qui attend jusqu'à ce qu'il soit conforme à la destination ou il peut envoyer le paquet au prochain groupe venant en sens inverse grâce au cluster co-directionnel [20].

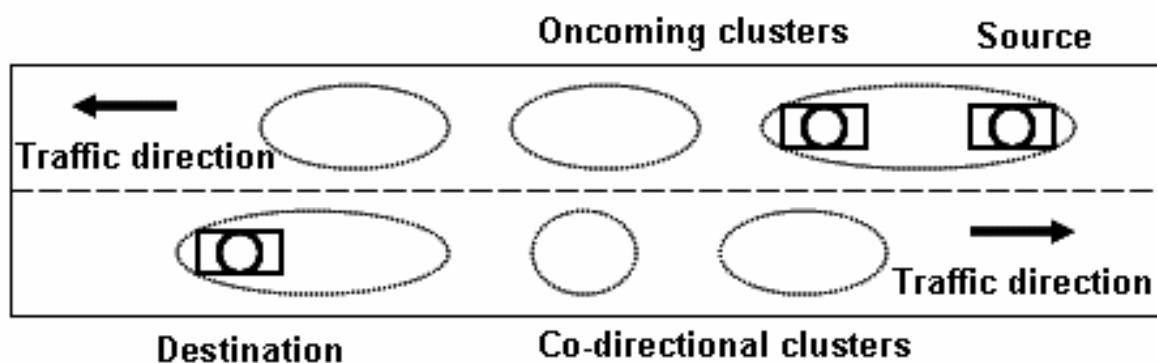


FIGURE 3.10 – de fonctionnement de TAPR[5]

3.6 (OMV) Overview of the proposed overlay multicast in VANET :

Cette approche se base sur une structure arborisante des nœuds du même groupe où on considère la racine comme source de flux multimédias qui sont inondés dans l'arbre d'une façon multicast(voir la Figure 3.11). le lien peut être entre un parent et son fils un multi-saut en passant par des nœuds qui ne sont pas intégrés dans l'arbre.

Dans un recouvrement d'un arbre, les paquets sont transmis en utilisant le protocole d'acheminement de couche réseau sous-jacent. Lorsqu'un nœud veut se joindre à l'arbre de multicast il demande en premier le BSN (la racine joue le rôle de nœud de bootstrap de l'arbre) pour lequel le nœud connaît son parent, de manière à devenir un membre de l'arbre. et pour assurer une bonne continuation pour la vidéo streaming sans coupure, il propose de lier chaque fils avec deux parents .

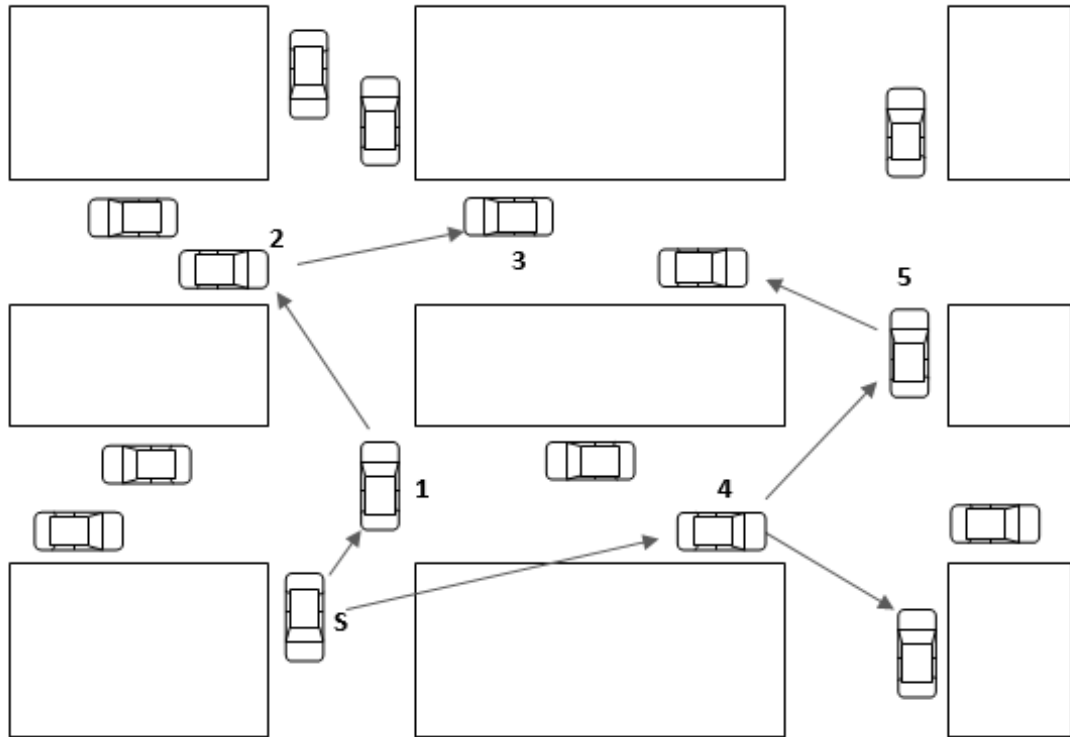


FIGURE 3.11 – exemple de multicast overlay dans un VANET urbain

3.7 Conclusion

Après la description des différentes approches qui sont proposés pour le VANET au but d'améliorer les performances de vidéo streaming par la diminution de nombre de paquet perdu et aussi le délai de transmission de bout en bout, pour cela dans notre travail on propose d'implémenter le protocole VAPER et faire des tests d'envoi des données vidéo a fin d'étudier les résultats qu'on obtiendra pour savoir a quel point on peut influer pour améliorer la transmission des données vidéo.

Chapitre 4

Simulation et Evaluation des résultats

4.1 Introduction

Afin d'évaluer la performance de l'une des approches discuté dans le chapitre précédant, nous avons choisi de passé par la simulation au lieu de l'expérimentation et la preuve mathématique spécialement à cause du coût de la première et la complexité de la deuxième, ainsi que la maîtrise de l'outil de simulation. Parmi les travaux discutés nous avons choisi d'évaluer les performances et les défis de la vidéo-streaming en utilisant le protocole VAPER. Nous commençons d'abord par une présentation des outils de simulation, puis les étapes de décomposition lors de l'envoi d'évaluer la QoS a la réception d'une vidéo. Enfin, nous discutons les différents scénarios, ainsi que les résultats obtenues.

4.2 Outil de simulation ns2 :

NS2 (Network Simulator) est un environnement de simulation des réseaux à événement discret origine du projet VINT (Virtual Inter-Network Testbed), qui fournit un support pour la simulation des différents protocoles des différentes couches. Le simulateur est basé sur le langage de commande Tcl (Tool Command Language) en tant que une interface de configuration, et le langage C++. Les fichiers de trace de simulation (.tr) et d'animation(.nam) sont créés au cours de la simulation. Une fois créés, les utilisateurs peuvent utiliser le fichier (.nam) pour vérifier le comportement à plusieurs reprises via

une animation graphique de réseau [21], et depuis le fichier de trace on peut facilement récupérer nos différents résultats de simulation. NS2 est le plus utilisé dans le domaine de la recherche pour ces raisons :

1. L'utilisateur peut concevoir et configurer le modèle de réseau à la fois à travers un générateur de scénarios ou par un code écrit à l'aide de langage (OTcl).
2. Il offre la possibilité d'émulation et de simulation rapide.
3. Il contient un nombre important des protocoles déjà mis en œuvre.
4. Le comportement dynamique peut être visualisé à l'aide d'un animateur.
5. Un logiciel Open source (code source libre) et extensible[22].

4.3 Génération de mobilité avec IMPORTANT :

Cet outil de génération de mobilité est utilisé pour générer des différents scénarios de mobilité utilisés pour évaluer la performance des protocoles de réseau ad hoc mobile. Elle nous offre trois modèles de maps : modèle du point de référence, Modèle de Freeway mobilité et modèle de Manhattan Mobilité.

Les fichiers de trace générés par cet outil sont compatibles avec le format requis par ns2. Ainsi, l'utilisateur pourrait intégrer directement les fichiers de trace générés par IMPORTANT dans le simulateur NS2 et réaliser ses simulations.

Dans notre étude, ce que nous intéresse c'est le modèle freeway, composé de quatre voies droites, deux dans chaque direction.

4.3.1 Modèle de Mobilité Freeway(FW) :

Ce modèle émule le comportement de mouvement des véhicules sur une autoroute. Il peut être utilisé dans le suivi de l'état de circulation ou le suivi d'un véhicule sur une autoroute. Caractéristiques importantes :

Dans ce modèle, nous utilisons des maps. Chaque map peut avoir plusieurs autoroutes et chaque autoroute à deux voies dans chaque direction. Les caractéristiques du modèle Freeway sont les suivantes :

1. Chaque véhicule est limité à sa voie sur l'autoroute.

2. La vitesse du véhicule est temporellement dépendante de sa vitesse précédente.
3. Si deux nœuds mobiles sur la même voie de l'autoroute sont à la distance de sécurité (SD), la vitesse du véhicule suivant ne peut pas dépasser la vitesse du véhicule précédent.

4.3.2 Définition d'une carte avec IMPORTANT :

Le fichier décrit ci-dessous indique la façon dans IMPORTANT pour créer un modèle d'autoroute avec quatre voies (on doit définir la direction, les coordonnées,..)

```

FREEWAY
FREEWAY_NUM 1
LANE_NUM 4
LANE_BEGIN 0 0 0 1 1
PHASE 0 (10.0,0.0) (10.0,1000.0) 18.0 25.0
LANE_BEGIN 0 1 1 1 1
PHASE 0 (20.0,0.0) (20.0,1000.0) 18.0 25.0
LANE_BEGIN 0 2 2 -1 1
PHASE 0 (30.0,1000.0) (30.0,0.0) 18.0 25.0
LANE_BEGIN 0 3 3 -1 1
PHASE 0 (40.0,1000.0) (40.0,0.0) 18.0 25.0

```

FIGURE 4.1 – exemple de fichier qui définit la structure de la carte

4.4 Vidéo simulator Evalvid :

EvalVid est un ensemble d'outils regroupés pour l'évaluation unifiée de la qualité de la transmission d'une vidéo. Il a une structure modulaire, ce qui permet l'échanger de la discrétion de l'utilisateur à la fois le système de transmission sous-jacente, et les codecs, il est applicable à tout type de système de codage, et peut être utilisé à la fois dans les vrais dispositifs expérimentaux ou dans des expériences de simulation. Les outils sont mis en œuvre dans ISO-C pour une portabilité maximale. Toutes les interactions avec le réseau se font via deux fichiers de trace. Il est donc très facile à intégrer EvalVid dans n'importe quel environnement de simulation [23].

4.4.1 Les exigences de base de l'outil :

Les vidéos prises doivent être de même taille (la taille de la matrice d'une image), de sorte que l'évaluation peut être faite correctement. Pour cela, un problème de dépendance entre le nombre d'images et la taille de la trace vidéo doit être résolu.

Les vidéos doivent être de même format, c'est-à-dire qu'à la fois la référence et la vidéo déformée doivent en format brut (EvalVid prend la vidéo uniquement au format brut).

Les deux fichiers des vidéos prises doivent utiliser le même codec.

4.4.2 La vidéo choisie pour la simulation :

En EvalVid il est nécessaire d'utiliser un fichier d'extension YUV. Les premières vidéos des exemplaires qui sont suggérées par l'UIT-T [24] sont : Foreman, Hall moniteur et Nouvelles. Ces vidéos sont acquises à partir d'un référentiel généralement utilisé pour des études d'évaluation de la qualité vidéo. Dans notre étude on a choisi la vidéo de Nouvelles « akiyo-cif » qui est caractérisé par les paramètres décrits dans le tableau(4.2) :

video	La taille	Langueur (s)	Résolution d'affichage	codec	Image par seconde
akiyo_cif	5.6M	10	QCIF (176×144)	24 bits RGB (RV24)	25

FIGURE 4.2 – Caractéristiques de la vidéo news

4.4.3 Les étapes de préparations de la vidéo pour la simulation :

Pour simuler une vidéo de ce type il faut obtenir ces quatre fichiers : a01.mp4, sd-a01, rd-a01, st-a01. En passant par les étapes suivantes :

- création du fichier.yuv par la commande `xvidencraw` : `xvid - encraw - iakiyo - cif.yuv - w352 - h288 - framerate30 - max - key - interval30 - oa01.264`
découper la vidéo en 30 frames de type pmp4 et créer a01.mp4 : `MP4Box - hint - mtu1024 - fps30 - adda01.m4va01.mp4`

- Pour l'évaluation de qualité de vidéo, il faut garder un fichier référence ref.yuv : `ffmpeg -i a01.mp4 a01-ref.yuv`
- on utilise l'outil `mp4trace` de EvalVid pour envoyer un fichier mp4 par le protocole RTP (Real time Transport Protocol) RTP / UDP à un hôte de destination spécifié : `mp4trace -f -s 127.0.0.1 12346 a01.mp4`
- pour avoir le fichier `.sdp` (Session Description Protocol), on utilise : `MP4Box -std -sdp a01.mp4 > a01.sdp`
- pour avoir le fichier trace `st` «sender trace» qui contient les informations de type des frames et la segmentation des paquets, la commande suivante est exécutée : `mp4trace -f -s 127.0.0.1 12346 a01.mp4 > st-a01`

1	I	16037	16	0.000
2	P	1173	2	0.034
3	P	1044	2	0.067
4	P	913	1	0.101
5	P	1240	2	0.134
.
.
.
118	H	182777	179	7.247
119	H	183003	179	7.427
120	H	183691	180	7.617
121	H	183495	180	7.809

FIGURE 4.3 – exemple de fichier sender trace

- pour faire une évaluation de qualité de la vidéo reçue, il faut avoir un fichier initial `sd` « sender dump » qu'on le compare avec le fichier reçu `rd` : `tcpdump -n -tt -v udp port 12346 > sd-a01`
- pour le fichier reçu `rd` « receiver dump » : `tcpdump -n -tt -v udp port 12346 > rd-a01`

4.5 L'évaluation du protocole VAPER :

Pour évaluer le protocole VAPER, il faut tout d'abord : définir des nouvelles structures de données dans le fichier header «myUdp.h » qui est utilisé dans les échanges des messages de contrôles, Définir la structure des paquets de données dans « paquet.h » ; implémenter les fonctions nécessaires dans un fichier source C++ « myUdp.cc ». Dans ce qui suit, on va expliquer le travail accompli afin d'intégrer le protocoleVAPER dans NS2.**Remarque :** vu la difficulté d'implémenter le protocole VAPER dans ns2 surtout en tenant compte de la nature dynamique de sa topologie, on a introduit des changements qui nous aident qui facilite certains aspects. **Les changements introduits :**

découper l'autoroute à des segments fixes. Dans chaque segment, le nœud le plus proche au centre est choisi comme ClusterHead. le nœud le plus éloigné de centre dans la direction du déplacement est choisi comme clusterTail. Le ClusterHead et le ClusterTail du premier segment sont choisis comme étant les nœuds qui peuvent nous connecter au point d'accès.

4.5.1 Le fichier myUdp.h :

C'est le fichier où on a déclaré la structure de données pour l'entête du protocole VAPER qui va supporter les données de contrôle nécessaires comme illustré dans la captures d'écran suivantes :

```

structhdr_vaper{
intMessageType;
int CHID;
intCHseg;
intposN;
intCHvois;
intdir;
inttypeN;
staticint offset_;// required by PacketHeaderManager
inline static int& offset() { return offset_;}
inline static hdr_vaper* access(const Packet*p) {
return (hdr_vaper*) p->access(offset_);
}
};

```

FIGURE 4.4 – L'entête des messages de contrôle dans VAPER

Dans notre protocole, on a défini une classe 'Tailtimer' pour gérer le temps d'attente avant de choisir unClusterTail. Et aussi on a besoin d'une autre classe 'CHtimer' pour gérer la diffusion périodique des messages de contrôle par le ClusterHead de la façon suivante.

```

classClusterTailtimer : public Handler {
public:
    ClusterTailtimer(myUdpAgent* ag, double p=1) : agent(ag), period(p) {
        p=1;
        started = 0;
        end = 0;
    }
    /*déclaration de ses fonctions */

    void start(void);
    void stop(void);
    voidsetPeriod(double p);
    voidsetVariance(double v);
    voidhandle(Event *e);
}

```

FIGURE 4.5 – déclaration de classe tailtimer

```

classClusterHeadtimer : public Handler {
public:
    ClusterHeadtimer(myUdpAgent* ag, double p=1) : agent(ag), period(p) {
        p=1;
        started = 0;
        end = 0;
    }
    /*déclaration de ses fonctions */

    void start(void);
    void stop(void);
    voidsetPeriod(double p);
    voidsetVariance(double v);
    voidhandle(Event *e);
    /*déclaration des variables */

    int         end;
    int         started;
    Event       intr;
    myUdpAgent *agent;
    double      period;
    double      variance;
};

```

FIGURE 4.6 – déclaration de CHtimer

4.5.2 Le fichier myUdp.cc :

Ce fichier contient toutes les opérations et les traitements de gestion des messages de contrôles, et aussi la circulation des paquets de données pour assurer un bon fonctionnement et une bonne réception des données.

4.5.3 Exporter la classe C++ en OTcl :

Pour permettre la création en OTcl d'un objet de la classe définie en C++, « myUdpAgent », il faut définir un objet de liaison « myUdpAgentClass » qui est dérivé de la classe

TclClass. Ceci est expliqué dans le schéma ci-dessous :

```
static class myUdpAgentClass : public TclClass {
public:
    myUdpAgentClass() : TclClass("Agent/myUDP") {}
    TObject* create(int, const char*const*) {
        return (new myUdpAgent());
    }
} class_myudp_agent;
```

FIGURE 4.7 – Format de l'objet de liaison

4.5.4 Exporter les variables C++ en OTcl :

Les variables membres de la classe définie en C++ peuvent être configurés en OTcl en leur appliquant la fonction d'association bidirectionnelle, qui crée une nouvelle variable membre de l'objet OTcl associé. La simulation utilise un script OTcl peut changer les valeurs des variables configurées et définies en C++. Les valeurs par défaut de ces variables sont définies dans le fichier OTcl « ns-2.26/tcl/lib/nsdefault.tcl » La figure suivante nous présente l'utilisation de la fonction « binding » qui fait la liaison entre les variables C++ et TCL :

```
myUdpAgent::myUdpAgent() : id_sd(0), openfile_sd(0),id_rd(0), openfile_rd(0), CTtimer(this),
CHtimer(this)
{
    | bind("packetSize_", &size_);
    bind("CHseg_", &CHseg_);
    bind("CHvois_", &CHvois_);
    bind("Nseg_", &Nseg_);
    bind("posN_", &posN_);
    bind("dir_", &dir_);
    bind("typeN", &typeN);
    IsClusterHead=0;
    clusterHead=0;
    clusterHeadAddr=0;
    CHseg_=0;           //.....daddi.....
    CHvois_=0;
    Nseg_=0;
    posN_=0;
    dir_=0;
    pos=0;
    typeN=0;
    ADRchvois=0;
    pkt_received=0;
    UdpAgent();
}
```

FIGURE 4.8 – déclaration de constructeur

4.6 Evaluation des performances :

On a validé la modification apporté à l'approche VAPER ; nous avons varié à chaque fois le nombre des véhicules qui circulent dans le réseau. Cette simulation nous a permis de générer les fichiers rd.txt qui contient les différentes informations lors de la réception, ce même fichier va être rediffusé sous le nom sd.txt qui contient : les numéros des paquets reçus, son ID et le temps de recevoir.

Après chaque obtention des résultats d'un scénario ; avec le script « etmp4 » de l'outil Evalvid on aura trois types de fichiers statistiques : Délai.txt, Loss.txt et rate.txt, avec lesquels on va d'évaluer le protocole. Délai.txt : contient le délai de transmission de chaque paquet de bout en bout. Loss.txt : contient le nombre de paquets perdus. Rate.txt : contient le taux moyen de transmission pour chaque paquet.

Après le calcul du moyen général de Délai de transmission pour chaque scénario, on a obtenu le graphe suivant :

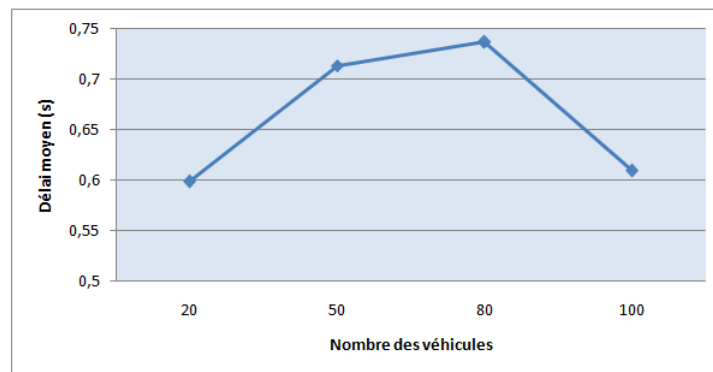


FIGURE 4.9 – évaluation de délai moyen en fonction de nombre de véhicule

D'après ces résultats obtenus, on peut dire que le délai de bout en bout dans notre cas n'a aucune relation avec le nombre de véhicules mais c'est surtout relié à la synchronisation entre les Cluster-Heads lors de la diffusion où l'inexistence de cette dernière peut conduire à plusieurs collisions donc retransmission qui va causer une augmentation de délai.

Après le calcul du moyen général des paquets perdus pour chaque scénario, on a obtenu le graphe suivant :

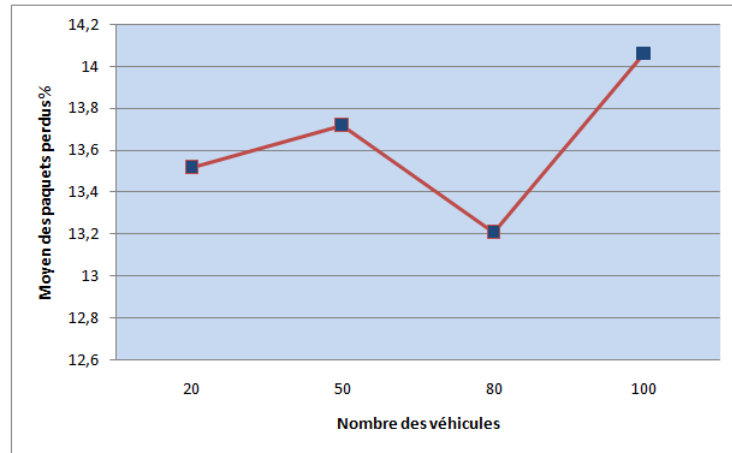


FIGURE 4.10 – évaluation de moyen des paquets perdus en fonction de nombre des véhicules

D'après ces résultats, on a remarqué que le pourcentage des paquets perdus s'augmente avec l'augmentation de nombre des véhicules à cause de densité de réseau où les véhicules sont assez proches ce qui peut conduire à plusieurs collisions donc retransmission qui va causer une augmentation de perte de paquet.

Après le calcul du taux moyen pour chaque scénario, on a obtenu le graphe suivant :

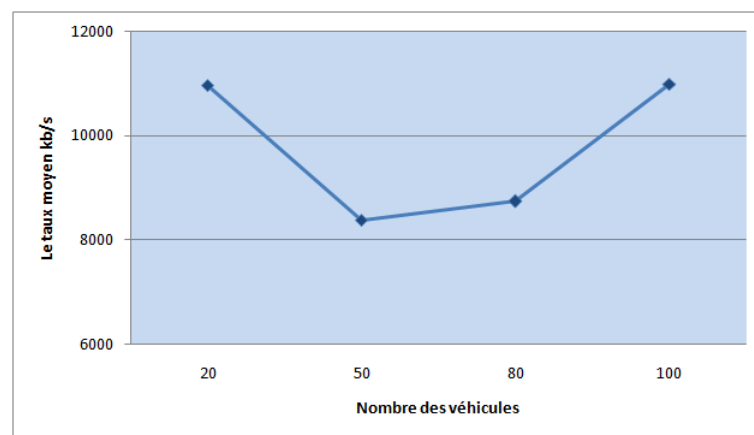


FIGURE 4.11 – évaluation de taux en fonction de nombre des véhicules

D'après ce graphe, on remarque que le taux du débit cumulé diminue pour un nombre de véhicules entre 20 et 50 et augmente pour un nombre entre 80 et 100. Cela prouve que le débit n'a aucune relation avec le nombre de véhicules, vu que les clusterHeads sont les seuls responsables de transmission et de la retransmission.

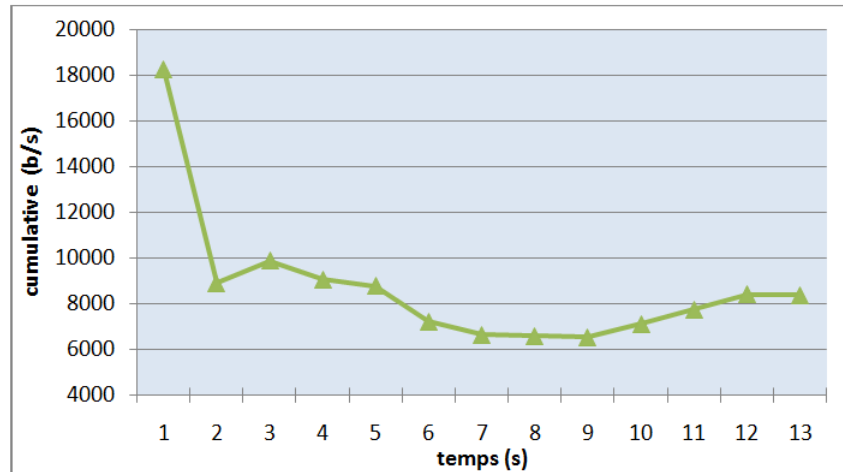


FIGURE 4.12 – évaluation de taux en fonction de nombre des véhicules

De la figure (4.12), on constate que le débit décroît pour stabiliser à 6 kb/s avant d'augmenter à nouveau, la diminution de débit s'explique par le fait que les paquets envoyés entrent en collision vu que chaque clusterHead est dans la portée de son (ses) voisin(s), ce qui génère un nombre important de collision.

4.7 Conclusion

Dans ce chapitre, on a simulé une version modifiée du protocole VAPER, les résultats obtenus ont montré une diminution du débit malgré que les seuls responsables de la transmission sont les clusterHeads, ce qui donne une idée sur la dégradation des performances dans les cas du problème de vidéo streaming dans un réseau dense où tous les nœuds sont des acteurs (des émetteurs, des récepteurs et des forwarders)

Conclusion et Perspectives

Les problèmes liés au développement des réseaux véhiculaires restent des défis ouverts, en particulier le problème de la forte mobilité et les obstacles

A cause de ces problèmes, le développement de protocoles pour la vidéo streaming est un objectif indépassable, pour assurer une bonne livraison des données tout en assurant les QoS qui sont essentiellement le délai court et un taux de perte réduit.

L'étude de quelques approches proposées dans ce domaine et l'implémentation de l'une de ces approches qui est le protocole VAPER nous a obligé de résoudre le problème de routage et la gestion des messages de contrôle, et nous a permis d'étudier l'efficacité du protocole dans le cas de transmission des données vidéo par l'évaluation de délai moyen de transmission et le taux des paquets perdus lors de transmission.

L'évaluation de ce protocole a été faite en utilisant le simulateur de réseaux NS-2, qui permet de valider les protocoles dans les conditions les plus proches à la réalité. Nous espérons que cette première étape va permettre aider ceux souhaitant travailler sur cet axe d'améliorer et proposer de nouveaux protocoles pour la vidéo streaming dans les VANets. Nous proposons comme perspectives futures de :

1. Redéfinir la procédure de création des clusters d'une façon plus dynamique, afin d'avoir de meilleures performances.
2. Réétudié le problème en intégrant d'autres paramètres comme : la taille de segment, la vitesse des nœuds variante.

Bibliographie

- [1] Communications véhicule à véhicule : applications et perspectives chapitre rédigé par rabah meraihi, sidi-mohammed senouci, djamal-eddine meddour et moez jerbi.
- [2] Dynamic overlay multicast for live multimedia streaming in urban vanets, yiling hsieh, kuochen wang, department of computer science, national chiao tung university, taiwan, computer networks 56 (2012) 3609–3628.
- [3] Codeplay : Live multimedia streaming in vanets, using symbol-level network coding, zhenyu yang, member, ieee, ming li, member, ieee, and wenjing lou, senior member, ieee, ieee transactions on wireless communications, vol. 11, no. 8, august 2012.
- [4] V3 : A vehicle-to-vehicle live video streaming, architecture meng guo, mostafa h. ammar, ellen w. zegura, networking and telecommunication group, college of computing, georgia institute of technology, 30339 atlanta, ga, united states, received 6 june 2005 ; accepted 22 august 2005, available online 28 september 2005.
- [5] Zipper : A zero-infrastructure peer-to-peer system for vanet, mahmoud abuelela, old dominion university, norfolk, virginia, usa, eabu@cs.odu.edu, stephan olariu, old dominion university, norfolk, virginia, usa, olariu@cs.odu.edu.
- [6] M.jerbi. protocoles pour les communications dans les reseaux de vehicules en environnement urbain. thèse de doctorat, universite d'evry val d'essonne, novembre 2008.
- [7] Un protocole de diffusion des messages dans les réseaux véhiculaires. mémoire de mr ahizoune ahmed, université de montréal.

- [8] Architecture pour communication véhicules-infrastructure, bertrand ducourthial* farah el ali* université de technologie de compiègne centre de recherches de royallieu laboratoire heudiasyc.
- [9] Un protocole de diffusion des messages dans les réseaux véhiculaires. mémoire de mr ahizoune ahmed, université de montréal.
- [10] Protocole de routage intelligent pour les réseaux ad hoc de véhicules mémoire présentée À l'université du québec en abitibi-témiscamingue par mohamed bouarir.
- [11] Routage dans les reseaux vehiculaires (vanet) cas d'un environnement type ville, présenté par : M. meraihi yassine, universite m'hamed bougara - boumerdes.
- [12] <http://www.learmstrong.com/dsrc/dsrchomeset.htm>.
- [13] Directional routing techniques in vanet, phd thesis, moath muayad al-doori, software technology research laboratory de montfort university leicester - united kingdom november 2011.
- [14] Le streaming dans les réseaux ad hoc, par hafnaoui taleb, université el hadj lakhdar batna algérie - master en informatique 2013.
- [15] Int. j. com. net. tech. 1, no. 2, 117-125 (2013), performance evaluation of adv with aadv for real-time and multimedia applications in vehicular ad-hoc networks (vanets), auteur : Omer masood, adeel akram, muhammad nadeem ma-jeed.
- [16] Master's thesis, high definition vidéo streaming using h.264 vidéo compression, author : Yassine bechqito, approved : 15.12.2009.
- [17] Master thesis electrical engineering october 2013, qoe of vidéo streaming over lte network, auteurs : Pradeep uppu, sushanth kadimpat.
- [18] Int. j. com. net. tech. 1, no. 2, 117-125 (2013), performance evaluation of adv with aadv for real-time and multimedia applications in vehicular ad-hoc networks (vanets), auteur : Omer masood, adeel akram, muhammad nadeem ma-jeed.
- [19] Delivering of live video streaming for vehicular, communication using peer-to-peer approach, yung-cheng chua and nen-fu huang alnstitute of communica-

tions engineering, national tsing hua university, taiwan, department of computer science, national tsing hua university, taiwan.

- [20] Poster : Traffic-adaptive packet relaying in vanet, mahmoud abuelela, old dominion university, norfolk, virginia, usa, eabu@cs.odu.edu, stephan olariu, old dominion university, norfolk, virginia, usa, olariu@cs.odu.edu.
- [21] M.a.rahman,a.pakštas,f.z.wang, "network modelling and simulation tools", simulation modelling practice and theory, vol.17,n.6, pp. 1011-1031,july 2009, issn 1569-190x.
- [22] S.v. mallapur, siddarama . r. patil"survey on simulation tools for mobile ad-hoc networks", iracst, international journal of computer networks and wireless communications (ijenwc), vol.2, n.2, april 2012.
- [23] Evaluation of video quality of experience using evalvid, venkataramana jonnalagadda vineelamusti, school of computing blekinge institute of technology 371 79 karlskrona sweden.
- [24] Itu-t recommendation p.910 : "subjective video quality assessment methods for multimedia applications", international telecommunication union, geneva, switzerland, 1996.