Titre: Title:	Contributions to Vehicular Communications Systems and Schemes
Auteur: Author:	Hazem Ahmed
Date:	2017
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Ahmed, H. (2017). Contributions to Vehicular Communications Systems and Schemes [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2729/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2729/
Directeurs de recherche: Advisors:	Samuel Pierre, & Alejandro Quintero
Programme: Program:	Génie informatique

UNIVERSITÉ DE MONTRÉAL

CONTRIBUTIONS TO VEHICULAR COMMUNICATIONS SYSTEMS AND SCHEMES

HAZEM AHMED
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

CONTRIBUTIONS TO VEHICULAR COMMUNICATIONS SYSTEMS AND SCHEMES

présentée par : AHMED Hazem

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

Mme BELLAÏCHE Martine, Ph. D., présidente

M. PIERRE Samuel, Ph. D., membre et directeur de recherche

M. QUINTERO Alejandro, Doctorat, membre et codirecteur de recherche

M. KHOMH Foutse, Ph. D., membre

M. KARMOUCH Ahmed, Ph. D., membre

DEDICATION

To my parents,

for your unconditional love and tireless support….

To my children,

my inspiration for success…

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Samuel Pierre and my co-supervisor Prof. Alejandro Quintero, not only for their guidance and time but also their support academically, emotionally and financially. I am grateful to Prof. Pierre, who gave me the opportunity to finish my research in his lab. His knowledge, professionalism, modesty, kindness and management style created a welcoming cradle of research and innovation. Special thanks to Prof. Quintero for being always available to me and being very supportive of my research. I will always remember the long discussion we had about life, work and research.

I would like to also thank all the jury members for their interest in my research and accepting to participate in the jury.

Special thanks to Mrs. Sabine Kebreau, the former lab manager, for her support in preparing for funding projects. Also, I would like to thank Mrs. Louise Longtin, Mrs. Micheline Lafrenière and Mrs. Brigitte Hayeur for clarifying the university policies.

Finally, I would like to thank all my colleagues in the lab for their valuable feedback on my presentations, especially Lamia for helping me with the French review.

RÉSUMÉ

La dernière décennie a marqué une grande hausse des applications véhiculaires comme une nouvelle source de revenus et un facteur de distinction dans l'industrie des véhicules. Ces applications véhiculaires sont classées en deux groupes : les applications de sécurité et les applications d'info divertissement. Le premier groupe inclue le changement intelligent de voie, l'avertissement de dangers de routes et la prévention coopérative de collision qui comprend la vidéo sur demande (VoD), la diffusion en direct, la diffusion de météo et de nouvelles et les jeux interactifs. Cependant, Il est à noter que d'une part, les applications véhiculaires d'info divertissement nécessitent une bande passante élevée et une latence relativement faible ; D'autre part, les applications de sécurité requièrent exigent un délai de bout en bout très bas et un canal de communication fiable pour la livraison des messages d'urgence.

Pour satisfaire le besoin en applications efficaces, les fabricants de véhicules ainsi que la communauté académique ont introduit plusieurs applications à l'intérieur de véhicule et entre véhicule et véhicule (V2V). Sauf que, l'infrastructure du réseau sans fil n'a pas été conçue pour gérer les applications de véhicules, en raison de la haute mobilité des véhicules, de l'imprévisibilité du comportement des conducteurs et des modèles de trafic dynamiques.

La relève est l'un des principaux défis des réseaux de véhicules, car la haute mobilité exige au réseau sans fil de faire la relève en un très court temps. De plus, l'imprévisibilité du comportement du conducteur cause l'échec des protocoles proactifs traditionnels de relève, car la prédiction du prochain routeur peut changer en fonction de la décision du conducteur. Aussi, le réseau de véhicules peut subir une mauvaise qualité de service dans les régions de relève en raison d'obstacles naturels, de véhicules de grande taille ou de mauvaises conditions météorologiques.

Cette thèse se concentre sur la relève dans l'environnement des véhicules et son effet sur les applications véhiculaires. Nous proposons des solutions pratiques pour les réseaux actuellement déployés, principalement les réseaux LTE, l'infrastructure véhicule à véhicule (V2V) ainsi que les outils efficaces d'émulateurs de relèves dans les réseaux véhiculaires.

Premièrement, nous avons introduit un protocole de relève qui est conçu pour gérer l'infrastructure basée sur la fibre optique. Cette recherche vise à modifier le protocole de relève actuel afin de bénéficier au maximum de l'utilisation de l'infrastructure PON (Passive Optical Network) pour réduire les frais généraux du réseau pendant la relève.

Deuxièmement, nous avons étudié les exigences d'un banc d'essai de véhicules adapté aux caractéristiques dynamiques des réseaux véhiculaires. Cette recherche démontre la nécessité d'un banc d'essai qui peut être un émulateur efficace pour les applications et les opérations véhiculaires tel que la relève. Ensuite, nous avons introduit une architecture de banc d'essai flexible qui peut être utilisée comme émulateur pour les applications de véhicules et les applications véhicule à véhicule (V2V). De plus, nous avons implémenté une application comme preuve de concept pour un scénario V2V.

Troisièmement, nous avons introduit une relève basée véhicules qui s'appuie sur la topologie du réseau routier. L'innovation consiste non seulement à utiliser des véhicules comme partenaires pour contribuer au processus de relève, mais aussi à augmenter le taux de réussite de la relève et la prévisibilité du prochain relais en utilisant la topologie du réseau routier.

Enfin, nous avons introduit un protocole de relève basé réseau qui est adapté aux réseaux de véhicules. Contrairement à la recherche précédente, ce protocole n'utilise pas de véhicules en tant que partenaires pour faciliter le processus de relève mais plutôt, la gestion de la relève dépend principalement du réseau. Le protocole nouvellement introduit peut être facilement intégré au réseau LTE et il peut gérer la nature complexe du réseau de véhicules. En plus du protocole innovant, nous avons introduit un modèle mathématique qui peut être utilisé pour analyser les futurs protocoles de transfert.

ABSTRACT

The last decade marked the rise of vehicular applications as a new source of revenue and a key differentiator in the vehicular industry. Vehicular Applications are classified into safety and infotainment applications. The former include smart lane change, road hazard warning, and cooperative collision avoidance; however, the latter include Video on Demand (VoD), live streaming, weather and news broadcast, and interactive games. On one hand, infotainment vehicular applications require high bandwidth and relatively low latency; on the other hand, safety applications requires a very low end to end delay and a reliable communication channel to deliver emergency messages.

To satisfy the thirst for practical applications, vehicle manufacturers along with research institutes introduced several in-vehicle and Vehicle to Vehicle (V2V) applications. However, the wireless network infrastructure was not designed to handle vehicular applications, due to the high mobility of vehicles, unpredictability of drivers' behavior, and dynamic traffic patterns.

Handoff is one of the main challenges of vehicular networks since the high mobility puts pressure on the wireless network to finish the handoff within a short period. Moreover, the unpredictability of driver behavior causes the traditional proactive handoff protocols to fail, since the prediction of the next router may change based on the driver's decision. Moreover, the vehicular network may suffer from bad Quality of Service (QoS) in the regions of handoff due to natural obstacles, large vehicles, or weather conditions.

This thesis focuses on the handoff on the vehicular environment and its effect on the vehicular applications. We consider practical solutions for the currently deployed networks mainly Long Term Evolution (LTE) networks, the Vehicle to Vehicle (V2V) infrastructure, and the tools that can be used effectively to emulate handoff on the vehicular networks.

Firstly, we introduced a handoff protocol that is designed to handle infrastructure based on fiber optics. This research aims at modifying the current handoff protocol to make a maximum profit of using the Passive Optical Network (PON) infrastructure to decrease network overhead during the handoff.

Secondly, we investigated the requirements for a vehicular testbed that is suitable for the dynamic features of the vehicular networks. This research clarifies the need for a testbed that can effectively emulate vehicular applications and operations (such as handoff). Then we introduced a flexible testbed architecture that can be used to emulate vehicular applications and V2V applications. Moreover, we implemented a proof of concept application for a V2V scenario.

Thirdly, we introduced a vehicular based handoff based on the road network topology. The innovation is not only using vehicles as partners to help in the handoff process but also increase the handoff success rate and the predictability of the next router by using the road network topology.

Finally, we introduced a network based handoff protocol that is tailored for vehicular networks. On the contrary of the previous research, it does not use vehicles as partners to assist in the handoff process and mainly depend on the network to handle the handoff. The newly introduced protocol can be easily integrated with the LTE network and can handle the complex nature of the vehicular network. In addition to the innovative protocol, we introduced a mathematical model that can be used to analyze future handoff protocols.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ANR	Automatic Neighbor Relation
AP	Access points
API	Application Program Interface
AR	Access Router
AWG	Arrayed Waveguide Grating
CAPEX	Capital expenditure
CDF	Cumulative distribution function
CoA	Care of Address
DAD	Duplicate Address Detection
DSRC	Dedicated Short Range Communications
EMA	Exponential Moving Average
EPON	Ethernet Passive Optical Network
GPS	Global Positioning System
HA	Home Agent
HD	High Definition
HoA	Home Address
IP	Internet Protocol
KPI	Key Performance Indicators
LBS	Location Based Services
LET	Link Expiration Time
LLID	Logical Link Identifier
LMA	Local Mobility Anchor
LTE	Logical Topology Emulation

MAG	Mobile Access Gateway
MANET	Mobile Ad hoc Network
MAP	Mobility Access Point
MIP	Mobile Internet Protocol
MME	Mobility Management Entity
MN	Mobile Node
MRO	Mobility Robustness Optimization
NAR	Next Access Router
NEMO	Network Mobility
OLT	Optical Line Terminal
ONU	Optical Network Unit
OPEX	Operational Expenditure
PAR	Previous Access Router
PBU	Publisher Binding Update
PK	Public Key
PL	Packet Loss
PMIP	Proxy Mobile Internet Protocol
PN	Partner Node
PO	Packet Overhead
PON	Passive Optical Network
PP	Permissibility Period
PtMP	Point to Multi-Point
PtP	Point to Point
QoE	Quality of Experience

QoS	Quality of Service
RO	Route Optimization
RSSI	Received Signal Strength Indicator
SDK	Software Development Kit
SDL	Smart Device Link
SDT	Service Disruption Time
SEDLANE	Simple Emulation of Delay and Loss for Ad Hoc Networks Environment
SIT	Service Interruption Time
SME	Shared Media Emulation
SNIR	Signal to Noise Ratio
SNR	Signal to Noise Ratio
SON	Self-Organized Network
SUMO	Simulation of urban mobility
SWINE	Simulator for Wireless Networks Emulation
TC	Traffic control
TDM	Time Division Multiplexing
TMC	Traffic Message Channel
TON	Transactions on Networking
TraNS	Traffic and Network Simulation
UE	User Entity
VANET	Vehicular Ad hoc Netowrk
VLET	Vehicle Link Expiration Time
VoD	Video on Demand
WMN	Wireless Mesh Networks

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

With the broad adoption of new software technologies in vehicles, vehicular applications are poised to be a promising market. VANET applications are classified into two types namely safety and infotainment. Safety and driver assistance applications are critical applications to enhance the road safety and mitigate road hazards, including cooperative collision avoidance, cooperative maneuvering, automated driving at low speed and on highways, and road condition detection [2]. However, the goal of infotainment applications is to enhance the driver and passenger experience by providing information and services such as interactive gaming, audio\video sharing, interactive messaging and news and weather information broadcasting.

The handoff problem is amplified in Vehicular Ad hoc Network (VANET) due to high mobility and the unpredictability of traffic, road conditions, and driver behavior. High mobility increases the number of networks that vehicles roam through and that increases the number of handoff requests on VANET and decreases the predictability of the next network. Moreover, high mobility adds more pressure on the vehicles to complete the handoff process in shorter time.

Modeling and analyzing the customer Quality of Experience (QoE) of vehicular applications and protocols are not easy tasks, since it involves emulating network communication with different protocols, fast movement and interactions on several road topologies with different speeds, and in the same time emulating the hardware to react to the user input similar to real experiment. The lack of such an emulator or testbed jeopardizes the development of VANET protocols and applications.

In this chapter, we will introduce the basic concepts and definitions used in this thesis, then the literature review followed by the thesis objectives and finally the thesis's outlines.

1.1 Basic Concepts and Definitions

This section presents to the reader the main concepts and definition that are used throughout this document. We will begin with the definition of Handoff since it is the anchor of this document. Subsequently, we will define the network based and partner based handoff. Afterwards, we will demonstrate the wireless backhauls, then wireless Handoff testbed, and finally the verification of handoff protocol main concepts will be defined.

1.1.1 Handoff

Handoff is the process of switching from a network or a base station to a neighboring one by a moving terminal. The moving terminal in the context of VANET is a vehicle. Without a proper handoff technique, the moving terminal may suffer from service degradation.

1.1.2 Network Based Handoff

Network based handoff is a type of handoff protocols that depends mainly on the network to perform the handoff tasks. This implies that the moving nodes in this protocol require minimum to none support for handoff functions. There are three popular network based handoff protocols: Proxy Mobile IP, Bicasting Based Handoff, and Forwarding Based Handoff.

A Proxy Mobile Internet Protocol (PMIP)

PMIP is a handoff protocol that depends mainly on two components the Mobility Access Gateway (MAG) and the Local Mobility Anchor (LMA) as shown in Figure 1.1. The first time a mobile attaches to a MAG, the MAG registers its address and notifies the LMA. Whenever the LMA receives data addressed to an MN; it looks up its internal table and forwards this data to the appropriate MAG, which in turns forward the data to the mobile node.

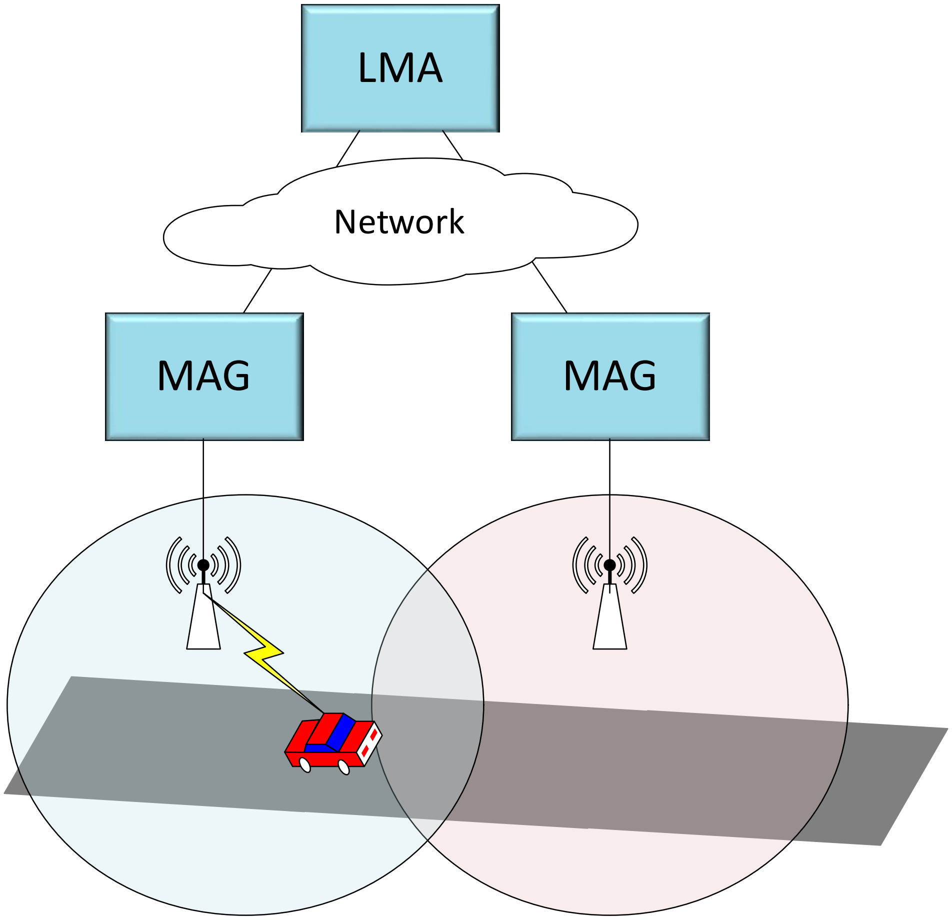


Figure 1.1: Proxy Mobile IP (PMIP)

B Bicasting Based Handoff

In the bicasting based handoff, to avoid packet loss network will bicast the data to both the old and new access routers. Although, this may cause duplicate data and network overheard, it minimizes the packet loss. It worth to mention that bicasting is not equivalent to duplicating the data since in bicasting the data is duplicated in an optimize manner based on the shared routes between the end nodes. In Figure 1.2, we show an example of data bicasting in an LTE network, when the data is received by the Service Gateway (S-GW) the data is bicasted to the Previous Access Router (PAR) and the Next Access Router (NAR).

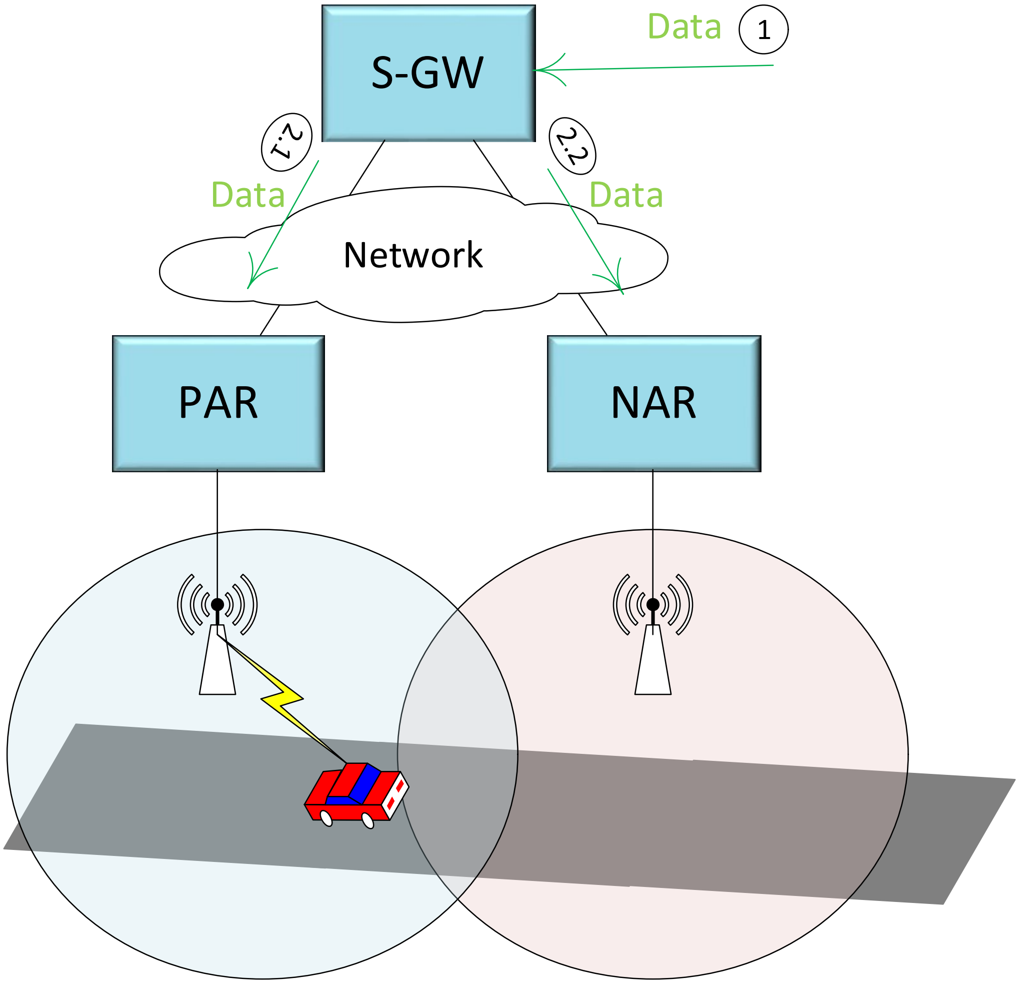


Figure 1.2: Bicasting in LTE

C Forwarding Based Handoff

To avoid packet loss, PAR sends all the data it receives, that is addressed to the roaming node, until the S-GW receives a notification that the handoff is completed. While the handoff once the PAR detects the mobile node is out of coverage, it starts forwarding the incoming data from PAR to NAR. At the NAR, the data will be buffered until the mobile node connects to the NAR where all the data will be delivered to it.

1.1.3 Partner Assisted Handoff

Partner assisted handoff depends on a partner node, such as a vehicle in the VANET context, to assist in the handoff process. The main idea of this protocol is the make use of other partners that have better network coverage to rely on handoff signals and/or mobile data to a roaming node. In Figure 1.3, we show an example of data forwarding on the LTE network.

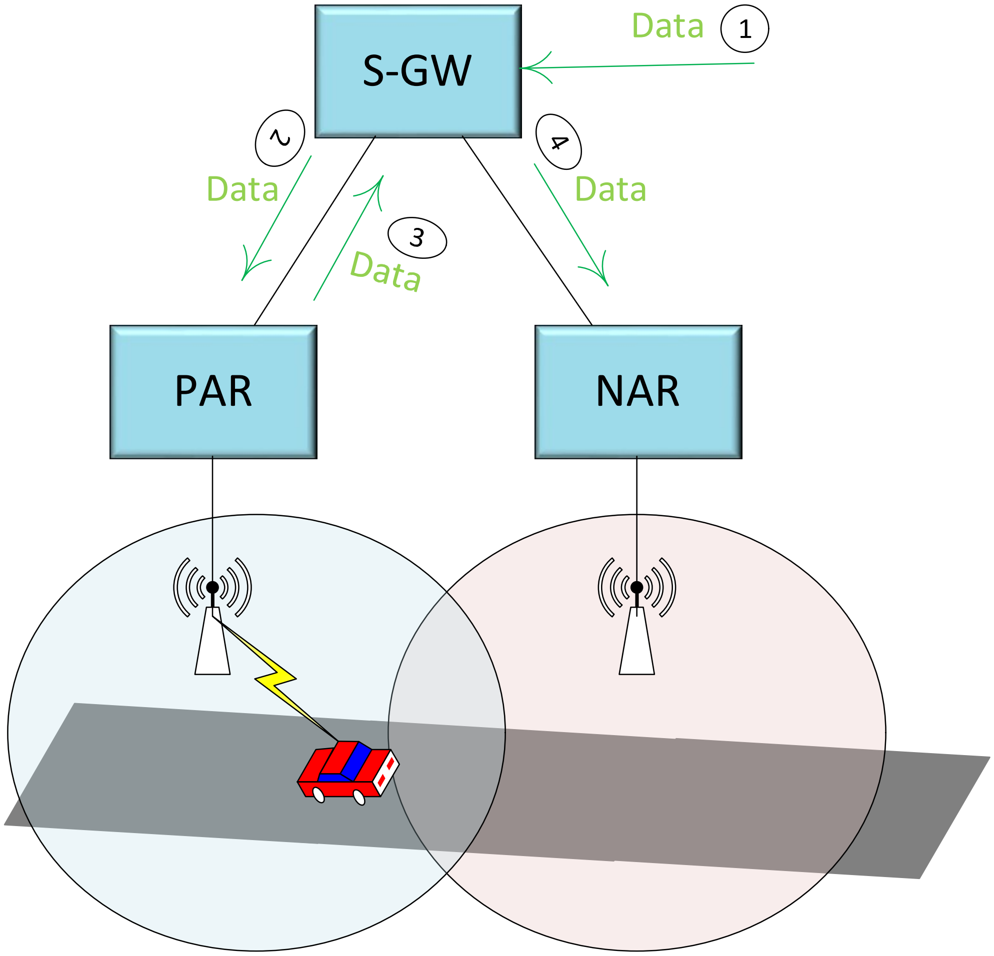


Figure 1.3: Data Forwarding in LTE

1.1.4 Self-Organized Networks

Self-Organized Networks (SONs) are the networks that their nodes can communicate to each other to enhance the network performance, heal from failures (self-healing) and optimize sharing network resources without human intervention [3]. SON is predicated to be a future research hot topic for next generation networks[4].

1.1.5 Wireless Backhaul

A Fiber Optics Networks

Fiber optics networks depend on light to transmit and receive data; therefore they do not suffer from interference from other electronics. The main advantage of using fiber optics is that it provides very high upstream and downstream bandwidth at a rate of Gb/sec with small signal attenuation that allows lines to be run up to 20 Km without the need to retransmit any signals[5].

B Ethernet Passive Optical Networks

The Ethernet Passive Optical Network (EPON) is considered passive since all the downstream data is passively broadcasted to the network terminals. EPON also uses using optical splitters that do not contain electronics, eliminating any electronic interference by nearby devices, overall resulting in an Operational Expenditure (OPEX) lower than other networks [5].

EPON provides very high upstream and downstream bandwidth at a rate of Gb/sec with small signal attenuation that allows lines to be run up to 20 Km without the need to retransmit any signals. Wireless Handoff Testbed

1.1.6 Verification of Handoff Protocols

There are several ways handoff protocols can be verified and analyzed. Although simulation and quality of service are usually used to analyze handoff protocols, they lack functional parameters that testbeds along with the quality of experience analysis provide.

A Quality of Service

The quality of Service (QoS) is a set of metrics aimed at measuring the performance of a protocol. In the case of a handoff, packet loss, bandwidth, service disruption time, and delay are usually used to evaluate handoff protocols.

B Quality of Experience

The quality of Experience (QoE) is a set of metrics that are aimed at measuring the user satisfaction of using a service or a technology [6]. QoE is related to the QoS since the user satisfaction maybe affected by QoS metrics. In the context of handoff, if a QoS metric such as packet delay is too much the user application may crash, and as a result, the QoE will be low.

C Hardware Based Testbed

Although simulation provides an easy way to evaluate large networks with several scenarios, it lacks the real-life scenarios and exceptions. Hence, using actual hardware to analyze and measure the performance of protocols is needed.

D Emulation Based Testbed

Using a hardware-only testbed can be very expensive to achieve. Since there are scenarios where verifying a network protocol is not practical, such as simulating partner selection in case of an accident or in a vast network of moving vehicles. The emulation based testbeds are combining both simulation and hardware to analyze and verify a system.

1.2 Literature Review

In this section, we will illustrate the literature review related to the handoff problems solved in this thesis. In the field of wireless handoff, there are two main types that we consider in this thesis namely: network and partner based mobility protocol; the former is depending on the core network to handle the mobility while the latter depends on the moving nodes. Moreover, the self-organized networks spurred as an important field of innovation for the future generation wireless networks, and since we designed a self-organized protocol, we will demonstrate the latest advancements in this area.

One of the challenges that we solve in this work is the unsuitability of the old generation wireless networks for the new generation handoff protocol. Since most of the wireless core networks are built on the Passive Optical Networks (PON), we highlight the literature review and the current advancement in this field. In Chapter 2, we introduced and innovative PON integration that is suitable for the new generation wireless network, especially handoff.

Lastly, we will highlight the literature for the wireless testbeds that are suitable for vehicular networks. There is a lack in the market for an efficient vehicle network testbed that can be used to emulate the complex interactions of the vehicle to vehicle communication. In Chapter 3, we study in depth all the available techniques and testbeds that can be used for the vehicular network.

1.2.1 Network Based Mobility Protocols

In network based mobility protocols, the handoff procedure is handled mostly by the network and with the minimum intervention of the mobile node. In this section, we will demonstrate the Proxy Mobile IP (PMIP) and the LTE based protocols. PMIP ,and its derivatives, is one of the most famous network based mobility protocols, while LTE based protocols are widely used in the currently deployed networks.

A Proxy Mobile IP

PMIPv6 can be effectively used with evolved 3rd Generation Partnership Project (3GPP) as shown in [7]. The authors propose a seamless mobility IP management protocol for evolved 3GPP. The proposed architecture shows the ability to perform handoff between 3GPP and non-3GPP networks with PMIPv6.

In [8], the authors conduct performance analysis of Proxy Fast Mobile IPv6 (PFMIPv6). The PFMIPv6 is mathematically analyzed against the standard PMIPv6, and the results show a better performance of PFMIPv6 in the highway scenarios. The research shows that mixing both reactive and proactive modes for PFMIPv6 gives the optimum results.

In [9], the authors provide an integration framework between geo-networking and PMIPv6. The authors integrate the single hop PMIPv6 to the European architecture standard to provide a multi-hop adaptation. The research proposes several scenarios using real mobility traces to evaluate the performance of Internet connectivity on Vehicle Adhoc Networks (VANET). However, in this research, the Mobility Access Gateway (MAG) is always stationary, and the case where the MAG is mobile is not addressed.

In [10], the authors carry out a mathematical analysis of PMIPv6 on the vehicular network with Network Mobile Basic Support (NEMO-BS). This integration shown in this research between PMIPv6 and NEMO-BS yields in performance in enhancements in terms of packet loss and QoS . The mathematical analyses demonstrated here have also been verified by real-life experiments and simulation.

B LTE and Bicasting Protocols

Reducing the signaling cost and the delay resulting from waiting for path switch in LTE-A networks is investigated in the literature. Guo et al. [11] introduce using a local mobility anchor to work as a proxy for incoming data before and after the handoff. Replacing the relatively far MME by a local anchor reduces the handoff delay. Moreover, it distributes the MME handoff message load on the local anchors.

The results show that using a local mobility anchor decreases the external handoff signaling cost dramatically [11]. However, using a proxy to relay data among neighbors introduces network overhead for not selecting the direct path between S-GW and eNB.

Lin et al. in [12] developed a mathematical model to compare the effect of location update on LTE-A networks. The authors analyzed three different techniques of location updates and their effect on the handover signaling cost.

Using bicasting for small and macro cells to enhance the handoff is discussed in [13]. The authors introduce a decoupled signaling, and user data plans architecture for the high-speed railway for 5G networks. In this architecture, the bicasted user and data plans are not sharing the same frequency bands. However, there is no optimization implemented to reduce the bicast period, and it assumed that the train could connect to both of eNBs at the same time.

To optimize the bicasting period, Dongwook et al. [14] suggest UE using velocity to reduce the bicasting period. By creating different speed profiles, each speed profile is associated with a Signal to Noise Ratio (SNIR) threshold. The results show a reduction in bicasting time. However, their protocol is relying on estimating the UE speed and selecting an accurate SNIR value which are not easy tasks and requires major modification to the original LTE protocol. Moreover, the bicasting is canceled after the handover is terminated and all communication is completed by MME. Hence, it may last longer based on the network delay between target eNB and MME.

Reactive Bicasting is used to reduce the Service Interruption Time (SIT) in handover between macrocells and femtocells in LTE networks in Guo et al.[15], the authors present a handover protocol where the bicast is initiated by MME once *HO Required* message is received instead of using the traditional forwarding. Since the bicasting starts too early in the handover process, there almost no packet loss and SIT is minimized. One benefit of this protocol is that it does not require any special messages or information from the UE to be shared with the MME.

1.2.2 Partner Assisted Handoff

In Partner assisted handoff mechanism HMIPv6 (P_HMIPv6)[16] Chen et al., propose a technique to reduce the handoff delay. The P_HIMPv6 uses one of the MNs in the future MAP domain to work as a Partner Node (PN), which is similar to IEEE 802.16J RS[16] and helps the MN to pre-perform the unfinished work of the L3 handoff. To select a PN by MN, PN should be static and located near to the MN. The localization proximity is measured using Received Signal Strength Indicator (RSSI) to decide if the relay node is near or not. The P_HIMPv6 is proven using mathematical analysis and simulation to reduce the handoff delay and improve the QoS. The methodology in P_HMIPv6 is similar to the FHMIPv6 in the sense that the L3 handling is performed before the actual roaming to the new region.

Using a PN to help in the handoff process is proven to enhance the handoff QoS [17], [18], [16]. However, the proposed optimum PN selection protocols in the literature are not suitable for VANET. Using RSSI is proven to result is selecting a wrong partner in MANET [18]. Since the fact that if a PN is near to an MN is not granting they will last in range until the handoff process is completed, mainly due to two reasons: the first is selecting a nearby PN that is moving in an opposite direction with a fast speed, as shown in Fig.5.1 the PN will be soon out of range. The second is selecting a PN which moves with much faster speed than the MN; this results in both MN and PN will not be in range soon. However, there might be other PNs in the range of MN that are moving almost with the same speed and will stay in range longer and got discarded, merely because they are not the nearest to the MN.

1.2.3 Self-Organized Networks

Self-Organized Network (SON) is an important feature of the future LTE networks, where eNBs can communicate to each other to enhance the network performance, heal from failures (self-healing) and optimize sharing network resources without human intervention [3].

Mobility Robustness Optimization (MRO) is one aspect of SON, where it aims at reducing the number of failed handover for LTE. As an application to MRO, Wegmann et al. [19] provide a protocol to reduce the handover failures in the inter-RAT environment. The Handover KPI parameters are collected from both RATs, to optimize the time to trigger the handoff. Automatic Neighbor Relation (ANR) is another feature of SON where an eNB can request UE to scan neighbor cell information and send it back. Using this feature eNBs can automatically update the neighbor information table and may get the connectivity information from the MME[3].

1.2.4 Passive Optical Networks

In order for EPON to be compliant with the Ethernet (802.3) standards, it must support Point to Point (PtP) and Point to Multi-Point (PtMP) modes. However, the PON does not naturally support the PtP architecture since its downstream is always in broadcast mode. To comply with this requirement, the 802.3ah and 802.3av standards provide a virtual PtP and PtMP connection, or Logical Topology Emulation (LTE). In the LTE, when data is sent from one ONU to another, it is broadcasted to all ONUs and discarded by all of them except for the one with the destination Logical Link Identifier (LLID). This operation is done seamlessly to the applications and services running on ONUs.

Other research relevant to the EPON architecture includes the development of a router through the use of Arrayed Waveguide Grating (AWG) [20] for Wave Division Multiplexing PON (WDM PON). This work enables the routing of separate optical wavelengths to separate ONUs. However, this research is not able to be applied practically yet due to several factors: additional work is needed to replace the splitter with the AWG router, the fact that it is quite expensive to deploy, its need for multiple channels receivers and transmitters in the central office, and its lack of transparency to the outside plant [21].

In [22], the authors introduce a solution for the lack of distributed communication support by using a hybrid tree-ring architecture for LTE backhauling. The innovation of this research is the enabling of the stand alone distributed control of the LTE as well as the enabling of direct communication between ONUs. However, it is not supported by the current PON systems and also results in extra overhead costs in the deployment of the network.

1.2.5 Wireless Testbed

Although simulation is a very efficient and economical technique to validate extensive simulations and hard to reproduce scenarios such as road accidents, it suffers from artificial models that are not always imitating the real environment. Also, some programs must run on the actual machine; this requires combining both real machines with large simulation.

Simple Emulation of Delay and Loss for Ad Hoc Networks Environment (SEDLANE) [23] is an extension for DummyNet in which NS-2 simulations are used as a source for delay and packet loss information that DummyNet requires operating. The idea is to use NS-2 data to simulate complex scenarios that would be tough, if not impossible, to replicate in real environments. Using network simulator outputs as an input for a link emulator allows for the simulation of complex traffic scenarios, obstacles and signal properties in different settings.

KauNet [24] can be considered an extension for DummyNet. It provides the ability to create traffic shaping profiles that can be data-driven and discrete data-driven for DummyNet. These configuration files ensure the reproducibility of the traffic shaping experience. Moreover, KauNet provides the ability to introduce bit errors, which is not supported by DummyNet. There is a slight performance difference between KauNet and DummyNet, an experiment performance analysis of which is studied in [25].

NISTNet[26] provides a robust traffic shaping tool. It is available in Linux kernel 2.4 and is patched to higher kernel versions. Similar to DummyNet and KauNet, NISTNet provides tools to throttle bandwidth, add packet delay and apply jitter. Moreover, it provides a GUI to configure traffic shaping parameters.

NetEM[27] and traffic control (TC) tools are widely used and integrated into Linux kernels. They are very similar in concept to NISTNet. However, NetEM and TC are superior in terms of the number of operating systems that support them. For example, Android framework currently supporting the TC and NetEM commands, along with most Linux distributions (with patching required).For an experiment-based performance analysis comparing DummyNet, NetEM, and NISTNet, refer to [28].

Conchon, E., et al. proposed W-NINE[29]: a two-stage emulation platform for mobile and wireless systems. Simulation for a preconfigured network with the nodes and their mobility can be conducted using Simulator for Wireless Networks Emulation (SWINE) to predict jitter, bandwidth, and other network characteristics. The output from the first stage is then used as input for a KauNet [24] controller router that applies these effects upon the nodes participating in an emulation. The idea of using two-stage simulation is time-consuming but very effective in reducing the time that a simulator needs to process the information. However, SWINE has not been updated, and its suitability for supporting and integrating with other road traffic mobility's simulators is not clear.

The idea of using multistage emulation is praiseworthy, but overall, a more generic platform that can integrate with different traffic and mobility simulators and utilize captured real life wireless traces.

1.3 Thesis Objectives

The main objective of this thesis is to propose models and algorithms for enhancing the handoff quality of service and quality of experience in vehicular ad-hoc networks. More specifically, this research aims at:

1. Investigates the problems of handoff caused by network infrastructure and propose a network-based mobility handoff protocol that is tailored for the new fiber optics networks;
2. Designing a self-organized network-based mobility protocol based on bicasting to improve the quality of service of handoff protocols;
3. Formulating a mathematical and analytical model that can be used to analyze the presented network-based mobility protocol and other future LTE protocols;
4. Studying the use of partners to enhance handoff on VANET and propose an effective VANET partner-based handoff protocol;
5. Present an architecture for a VANET testbed for verifying and analyzing the QoE of applications and handoff protocols;
6. Conducting a performance analysis of the proposed algorithms and protocols.

1.4 Thesis Outline

The rest of this thesis is organized as follows:

Chapter 2, introduces a network-based mobility management protocol that is tailored for Passive Optical Networks (PON) infrastructure. The new protocol makes use of the downstream property of the PON infrastructure to reduce the packet overhead whenever there is a handoff. The proposed protocol is based on Proxy Mobile IP (PMIP). However the analysis showed that the original PMIP suffers from network overhead and the proposed protocol effectively reduce packet overhead in case of handoff.

Chapter 3, discusses the validation and verification of VANET applications. Although simulation is widely used for validating wireless network applications, there is a lack for a testbed that is suitable for VANET. In this chapter, we compare, in details, the currently available simulators for both network and mobility based applications. Also, we compare the emulators and analyze their suitability for VANET applications. Finally, we draw a conclusion with the current challenges of creating a VANET testbed.

Chapter 4, introduces a flexible testbed that is tailored for VANET applications. The proposed testbed is not coupled to any specific mobility simulator and can be utilized with any network simulator. Hence it is flexible. The introduced testbed is suitable for emulating a wide range of VANET applications including V2V and handoff ones. We showed an example implementation with scenarios of a vehicle moving in a downtown environment and communicating with an Access Point (AP) and another vehicle.

Chapter 5, presents a novel a cooperative partner based handoff protocol. The introduced protocol makes use of road topology and maps to enhance the handoff on VANET. Also, it makes use of partner vehicle to assist in handoff and improve the handoff success rate. The results show the superiority of the proposed protocol as compared to other partner based mobility management protocols.

Chapter 6, investigates the LTE handoff thoroughly and introduces an innovative self-organized handoff protocol. The proposed protocol makes use of the delay of previous and next networks to adjust the handoff parameters to ensure a better QoS. Moreover, we introduce a detailed mathematical framework that can be used to analyze future LTE-based network protocols.

Chapter 7, introduces an innovative method for vehicular network to increase the reliability of vehicular networks by tackling the problem of dead spots. Comparative analysis with the state of the art inventions and protocols is held. The results show the suitability of the proposed protocol to the VANET.

Chapter 8, presents the general discussions of the this thesis. Finally, chapter 9 draws the contribution, research limitations and the direction of the future work.

CHAPTER 2 ARTICLE 1: NETWORK-BASED MOBILITY MANAGEMENT PROTOCOL FOR ETHERNET PASSIVE OPTICAL NETWORKS

Hazem Ahmed
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
hazem.ahmed@polymtl.ca

Samuel Pierre
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
samuel.pierre@polymtl.ca

Alejandro Quintero
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
alejandro.quintero@polymtl.ca

Published in Proceedings of the third ACM international symposium on design and analysis of intelligent vehicular networks and applications. ACM, 2013.

ABSTRACT

With increasing demand on bandwidth and mobile media applications, pressures upon network providers to provide more bandwidth to users and better Quality of Service (QoS) is increasing. For example, High Definition (HD) Video on Demand (VoD) streaming requires an average of 10 Mb/sec per user. Meeting these needs through the use of fiber optic networks to backhaul the traffic is essential, since fiber optic networks provide bandwidth measured at a rate of Gb/sec. However, most mobility management protocols are not built to handle fiber optic networks. Ethernet Passive Optical Network (EPON) is the dominant fiber optic network in the market due to its simplicity and cost of deployment. This paper describes how EPON can be used effectively to meet the need for increased bandwidth and handoff demands of mobile terminals. This paper studies Proxy Mobile IPv6 (PMIPv6) as the dominant network-based localized mobility management (Netlmm) protocol. Vehicle Ad Hoc Network (VANET) amplifies the handoff problems due to its unique characteristics. Hence, the issues and challenges of using the EPON network with PMIPv6 will be analyzed. Finally, recommendations will be established to optimize the usage of EPON with local mobility management protocols. This research shows by means of analysis the need for new protocols to support fast mobile IP optimization and then propose a solution that is tailored for EPON and analyze the proposed solution mathematically.

2.1 Introduction

With the increasing demand for bandwidth for mobile multi-media applications, it is estimated that the average household bandwidth requirement will jump to 1Gb/sec by 2020 [22]. In providing this increased bandwidth, network providers are interested in keeping their Operational Expenditure (OPEX) low and reaping the highest benefits from existing or new infrastructure with minimum capital expenditure (CAPEX).

In order to keep CAPEX and OPEX low in a system while at the same time increasing bandwidth requirements, fiber optic networks are used. Ethernet Passive Optical Networks (EPON) is a dominant fiber optic access network that easily meets demands for increased bandwidth and Quality of Service (QoS) [5]. EPON provides the best solution for minimizing CAPEX and OPEX, as compared to all the available wired and fiber technologies, and has the advantage of higher bandwidth [30].

The handoff problem is amplified in VANET due to high mobility and the unpredictability of traffic, road conditions, and driver behavior. High mobility increases the number of networks that vehicles roam though and that increases the number of handoff requests on VANET and decreases the predictability of the next network. Moreover, high mobility adds more pressure on the vehicles to complete the handoff process in shorter time.

Proxy Mobile IPv6 (PMIPv6) [31] is a dominant Netlmm protocol for handling mobility. The major advantage of using PMIPv6 as a network mobility management protocol lies in its ability to handle mobility with the mobile node interaction. This saves mobile node power, reduces the wireless usage, and makes the mobility management protocol transparent to the consumer. PMIPv6 and its extensions will be discussed in further detail in Section 3.

The combination of PMIPv6 with an EPON network is a solution to demands for increased bandwidth because fiber optics provides the highest reliability, the lowest cost per bit, and has the ability to cover large geographical areas [5, 30].However, current network mobility management protocols are not designed to optimize the usage of optical networks between major entities. This increases both the signaling cost and network overhead.

The remainder of this paper is organized as follows: Section 2.2 explains the EPON architecture. Section 2.3 demonstrates the uses of PMIPv6 and its related extensions. Section 2.4 describes the challenges and issues with PMIPv6 and its extensions when used in conjunction with EPON. Section 2.5 demonstrates the proposed solution and Section 2.6 explains mathematically the packet overhead of the proposed solution compared to traditional fast mobile protocols. Finally, Section 2.7 draws conclusions and the future work.

2.2 Ethernet Passive Optical Networks

This section gives an overview of EPON and its benefits, as well as presenting some of the latest advancements in EPON development.

2.2.1 EPON Architecture

Optical networks depend on light to transmit and receive data; therefore they do not suffer from interference from other electronics. EPON provides very high upstream and downstream bandwidth at a rate of Gb/sec with small signal attenuation that allows lines to be run up to 20 Km without the need to retransmit any signals. EPON also uses using optical splitters that do not contain electronics, eliminating any electronic interference by nearby devices, overall resulting in an Operational Expenditure (OPEX) lower than other networks [5].

Within the EPON architecture there are three major components: the Optical Line Terminal (OLT), the Optical Network Unit (ONU), and the optical splitter. The Passive Optical Network (PON) is considered passive since all the downstream data from the OLT is passively broadcasted to the ONUs by the optical splitter, as shown in Figure 2.1.

Standard EPON (1G-EPON) technology provides 1Gb/sec connections for upstream and downstream data transfer. However, with the recent development of 10G-EPON technology, a net throughput of up to 8.71 Gb/sec can be achieved [32]. As EPON becomes more developed, it is going in the direction of increased bandwidths while keeping the same architecture of downstream multicast using a splitter, due to its simplicity and cost effectiveness.

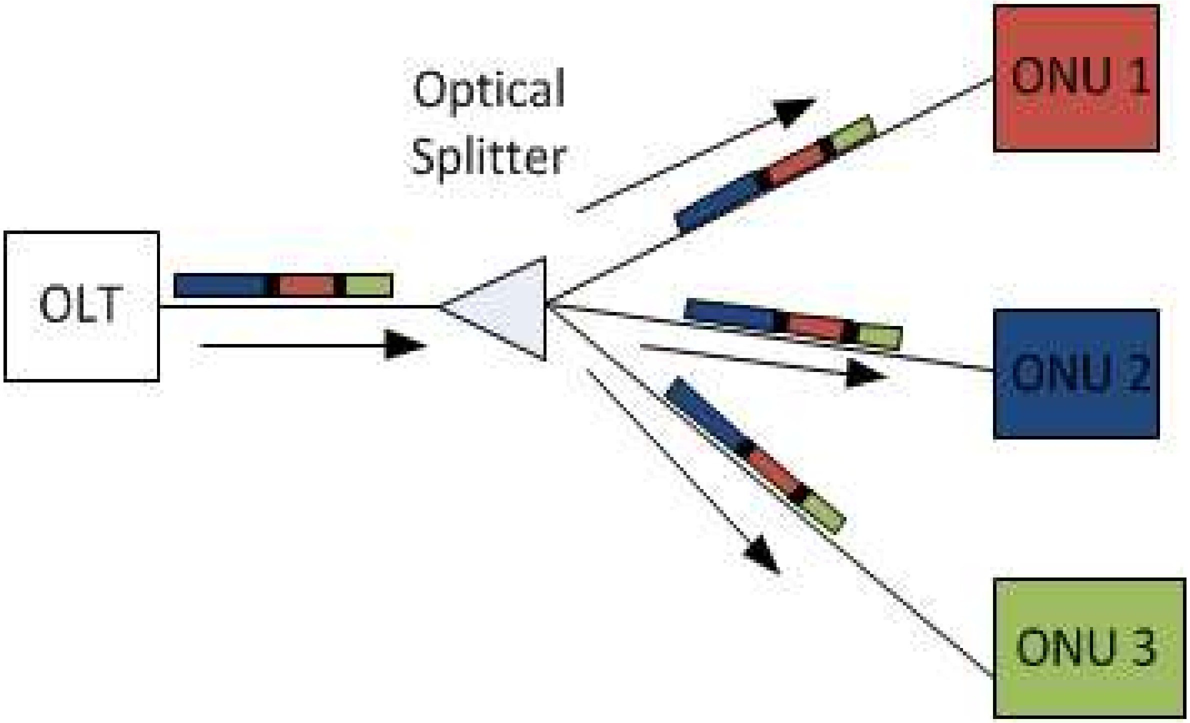


Figure 2.1: EPON Downstream

The upstream traffic is Time Division Multiplexing (TDM) based. This enables each ONU to communicate with the OLT directly. There is no need to provide collision avoidance procedures on the PON upstream, due to the synchronization mechanism used by the technology.

One critical aspect to consider when selecting a backend network backhaul for mobility management is energy consumption. PON outperforms all other fiber options for point to point, wireless, and wired networks when it comes to maintaining a low energy consumption for each transmitted bit [33].

2.2.2 Latest Advancements in EPON

In order for EPON to be compliant with 802.3 standards, it must support Point to Point (PtP) and Point to Multi-Point (PtMP) modes. However, the PON does not naturally support the PtP architecture since its downstream is always in broadcast mode. To comply with this requirement, the 802.3ah and 802.3av standards provide a virtual PtP and PtMP connection, or Logical Topology Emulation (LTE). In the LTE, when data is sent from one ONU to another, it is broadcasted to all ONUs and discarded by all of them except for the one with the destination Logical Link Identifier (LLID). This operation is done seamlessly to the applications and services running on ONUs.

Other research relevant to the EPON architecture includes the development of a router through the use of Arrayed Waveguide Grating (AWG) [20] for Wave Division Multiplexing PON (WDM PON). This work enables the routing of separate optical wavelengths to separate ONUs. However, this research is not able to be applied practically yet due to several factors: additional work is needed to replace the splitter with the AWG router, the fact that it is quite expensive to deploy, its need for multiple channels receivers and transmitters in the central office, and its lack of transparency to the outside plant [21].

In [22], the authors introduce a solution for the lack of distributed communication support by using a hybrid tree-ring architecture for LTE backhauling. The innovation of this research is the enabling of the stand alone distributed control of the LTE as well as the enabling of direct communication between ONUs. However, it is not supported by the current PON systems and also results in extra overhead costs in the deployment of the network.

In [30], the authors study the site planning for the use of PON as a wireless backhaul for cooperative wireless networks. They propose a new linear optimization based system to help in the selection of optimum placement of base stations and infrastructure. The results show an improvement in the cost and scalability of the system.

In [34], the authors provide a novel technique for hybrid wireless access networks. The proposed technique introduces a new hierarchical frame aggregation backhauled by fiber networking that works on Wireless Mesh Networks (WMN). The authors use two frame aggregation techniques: Aggregate MAC Protocol Data Unit (A-MPDU) and Aggregate MAC Service Data Unit (A-MSDU). Simulation, mathematical analysis and experiments show the enhancement network throughput and reduction of packet loss and jitter when using the frame aggregation technique.

2.3 **Proxy Mobile IPv6**

(PMIPv6) [31] is an Netlmm that is used to enable network mobility with mobile terminal interaction. Network providers generally prefer Netlmm, so the network operator can change the protocol transparently to the consumer and without any changes to the mobile terminals.

The major benefit of the PMIPv6 is that a mobile terminal can have just an IPv6 stack with no mobility support and it will still work. Since the user does not handle the mobility, this eliminates the mobility management messages between the user and the serving access router, which saves on both wireless bandwidth and mobile terminal power.

2.3.1 PMIP Architecture

PMIPv6 contains two major entities: the Local Mobility Anchor (LMA) and the Mobility Access Gateway (MAG), as shown in Figure 2.2. The LMA is the first point of contact for any data sent to a Mobile Node (MN). The LMA advertises the Care of Address (CoA) of the node and contains a mapping between the MAGs and each of the attached mobile nodes to it.

MAG works as an access router for the mobile nodes. It has the ability to translate the CoA to the Home Address (HoA) of the mobile node that is used to communicate with the mobile node.

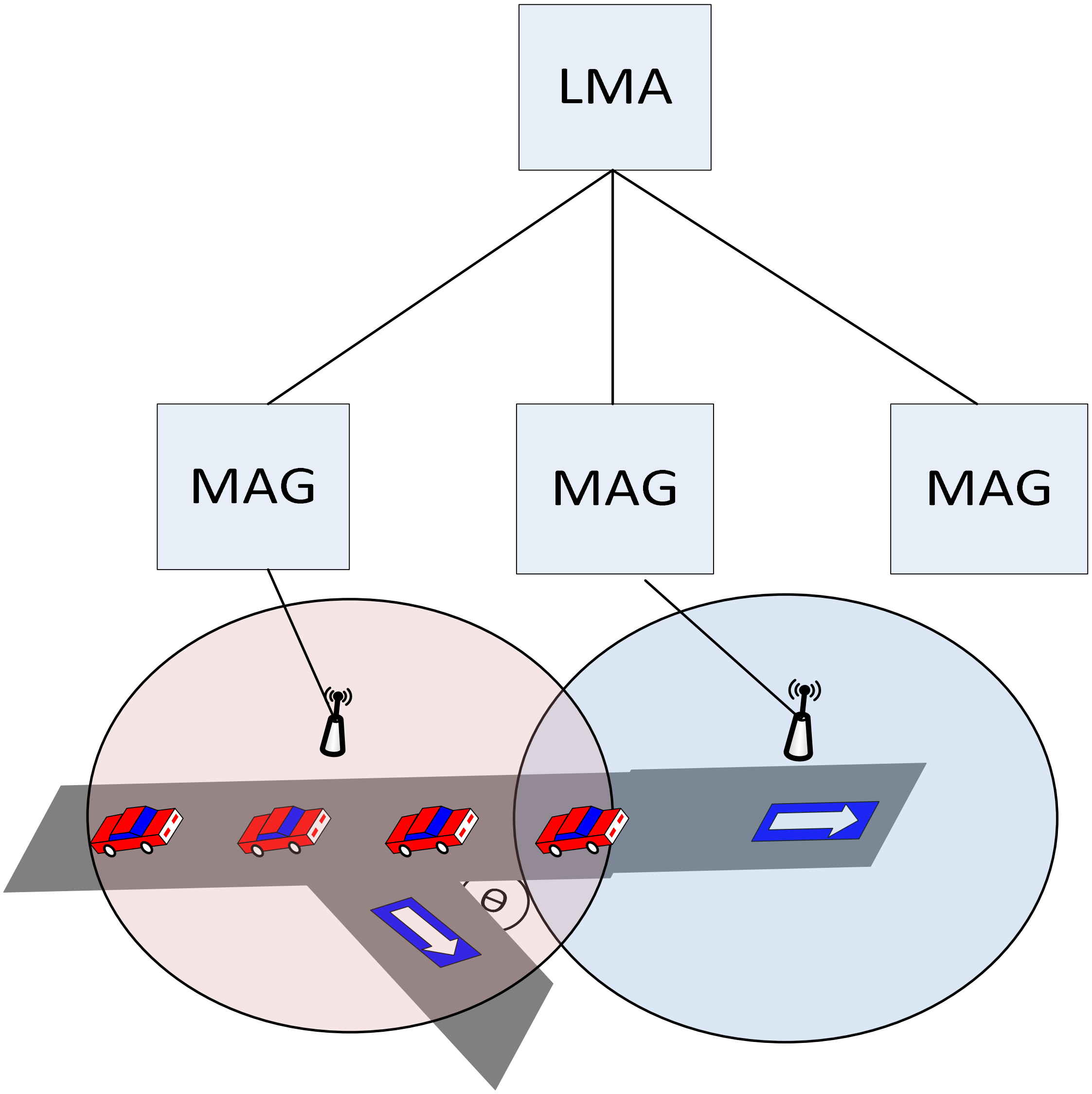


Figure 2.2 : PMIPv6 Architecture

Figure 2.3 shows the typical PMIPv6 protocol steps. It should be noted that this is the standard PMIPv6 without the use of EPON as an underlying network. The mobile station is exchanging router soliciting and router advertisement with the MAG1. The MAG1 then sends a Publisher Binding Update (PBU) to the LMA with the mobile node identifier. The LMA performs the required security checks and determines if the mobile node is granted access. After that, the LMA updates its lookup table to attach this MN to this MAG. A Publisher Binding Acknowledgment message is sent to MAG with the created CoA. The MAG1 then updates its lookup table with the mapping between the HoA and CoA.

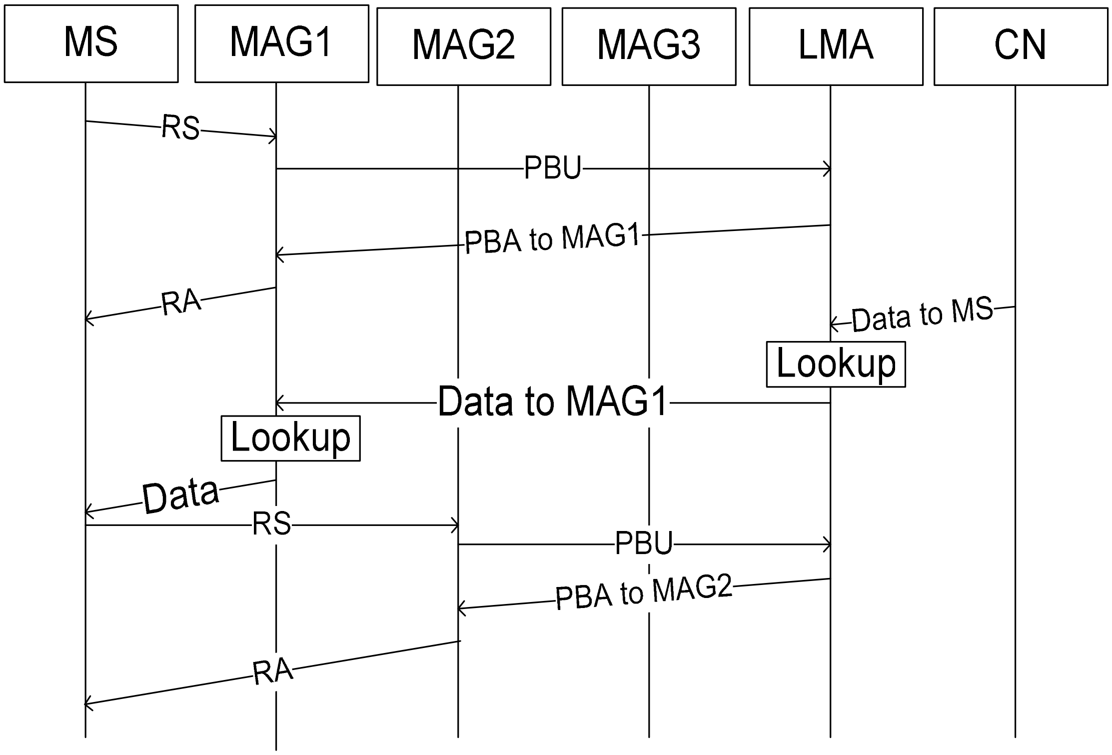


Figure 2.3: Typical PMIPv6 Flow Diagram without EPON

2.3.2 Advancements in PMIPv6

PMIPv6 can be effectively used with evolved 3rd Generation Partnership Project (3GPP) as shown in [7]. The authors propose a seamless mobility IP management protocol for evolved 3GPP. The proposed architecture shows the ability of performing handoff between 3GPP and non-3GPP networks with PMIPv6.

In [8], the authors conduct performance analysis of Proxy Fast Mobile IPv6 (PFMIPv6). The PFMIPv6 is mathematically analyzed against the standard PMIPv6 and the results show a better performance of PFMIPv6 in the highway scenarios. The research shows that mixing both reactive and proactive modes for PFMIPv6 gives the optimum results.

In [9], the authors provide an integration framework between geo-networking and PMIPv6. The authors integrate the single hop PMIPv6 to the European architecture standard to provide a multi-hop adaptation. The research proposes several scenarios using real mobility traces to evaluate the performance of Internet connectivity on Vehicle Adhoc Networks (VANET). However, in this research, the Mobility Access Gateway (MAG) is always stationary and the case where the MAG is mobile is not addressed.

In [10], the authors carry out a mathematical analysis of PMIPv6 on vehicular network with Network Mobile Basic Support (NEMO-BS). This integration shown in this research between PMIPv6 and NEMO-BS yields in performance in enhancements in terms of packet loss and QoS . The mathematical analyses demonstrated here have also been verified by real-life experiments and simulation.

2.4 Issues and Challenges

The downstream broadcasting and the lack of native PtP connection amongst ONUs disturb the performance and throughput of traditional NBM management.

The basic PMIPv6 and EPON integration and data flows are shown in Figure 2.4 and Figure 2.5. Figure 2.5 shows an implementation of the system where the LMA is placed at the OLT and the MAGs are connected to the ONUs. Note that the EPON specification describes a Virtual ONU, which enables the addition of more than one ONU entity on the same ONU hardware.

Figure 2.5 demonstrates the data plan of the data receipt operation. When data is sent to a mobile station, the LMA receives the data first and tries to find the MAG that has this MS connected to it. Since the LMA is placed at the OLT, the data is broadcasted to all the MAGs. However, only the MAG with the LLID will process the packet.

Since the PMIPv6 does not take into consideration this requirement that data be broadcast to all ONUs, this could result in some performance issues. This section explores localized routing for PMIPv6, the tunneling effect, and fast mobile IP support. These extensions were selected for consideration because they are the most widely used of all PMIPv6 extensions. Furthermore, this analysis will only consider the data plan, due to its obvious effect on the network throughput and bandwidth.

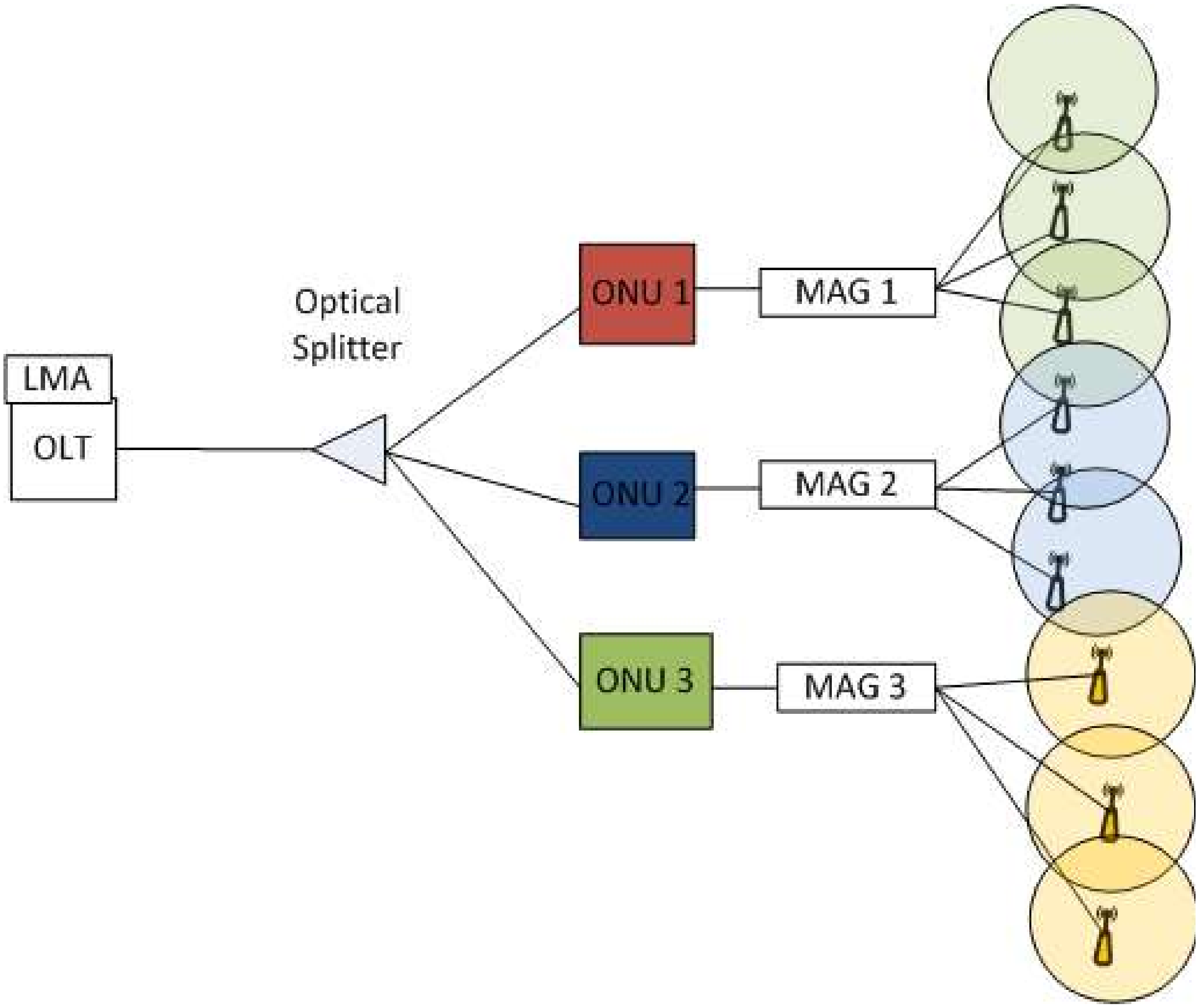


Figure 2.4: PMIPv6 on EPON

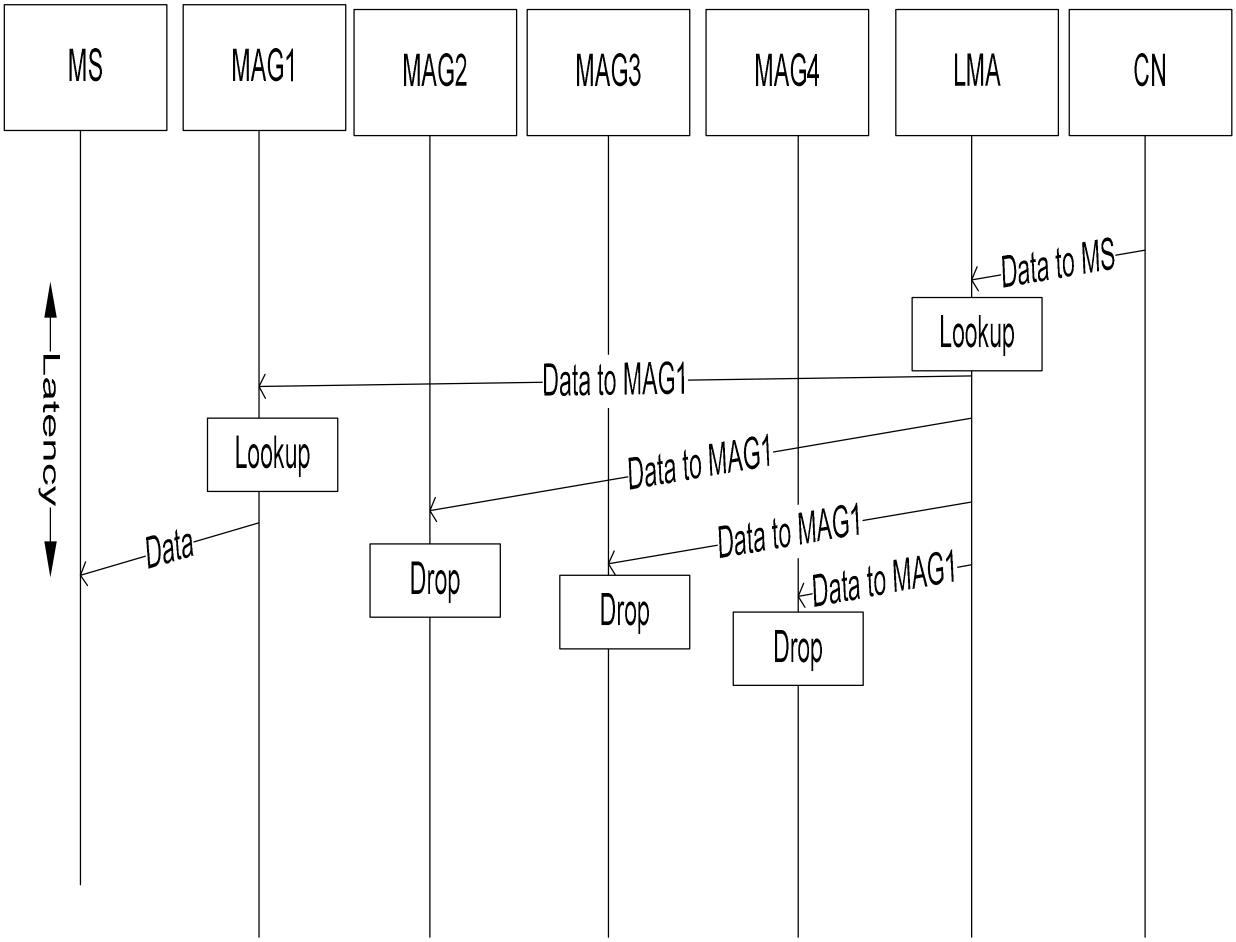


Figure 2.5: Data Plan Flow of PMIPv6 with EPON

2.4.1 Tunneling between LMA and MAG

The PMIPv6 standard requires the use of a tunnel between the LMA and each MAG. When the LMA receives an external message addressed to an MN with its CoA, the LMA will then perform a look-up operation to find the corresponding MAG with that particular MN. The LMA will encapsulate the received packet into another packet addressed to the MAG, and then the MAG will de-capsulate the message before translating the CoA address to the HoA of the MN and finally sending the data to the MN.

This technique is used to ensure that the data packet from the LMA arrives at its destination on the MAG. However, since there is no multi-hop communication in the PON between the OLT and

ONUs, tunneling is not needed for this process. The LLID of the MAG can be used to send the packet from LMA to the MAG so that the MAG can subsequently use the CoA in an Ethernet packet to find its HoA and send the received data to the MN.

2.4.2 Fast Mobile IP Protocols

Fast Mobile IP protocols are generally based on buffering data on either the new or old access router during handoff. When the connection to the new router is completed, the buffer data is then sent to the MN. There are multiple introduced protocols and extensions to enable this fast mobile IP. FMIPv6 [35] is one of the first attempts to enable this function on the Mobile IPv6. The Fast Handovers for Proxy Mobile IPv6 standard [36] is used to enable fast handovers for PMIPv6.

The main benefit of using fast mobile protocol is the avoidance of packet loss because all the traffic is buffered while the node is unreachable. However, the fast mobile protocol requires access routers or MAGs with large buffer capabilities, though it may suffer from buffer overflow and delays in packet delivery.

In the EPON architecture, the overhead of implementing fast mobile IP protocol is exceptionally high. As shown in Figure 2.5 since the PtP communication is done through the LMA (i.e. the OLT), all data will be broadcasted to every MAG (i.e. the ONUs), and only the MAG with the specified LLID will process the packet, while all the other MAGs will drop it.

With bandwidth intensive applications such as VoD or multi-media streaming, network throughput could be dramatically degraded. In Figure 2.5, if the handoff is done between two MAGs under the same LMA (e.g. MAG1 and MAG2) then the data that is buffered is sent to all MAGs (including MAG1 and MAG2). It will then be buffered on MAG1 only, and then sent to MAG2. Sending the data from MAG1 to MAG2 will trigger a broadcast of the data to all MAGs where all of them will drop the packets except MAG2.

In [37], the authors introduce a VoD service architecture for EPON networks. This research highlights the challenge of requesting the same video from different ONUs attached to the same OLT. In this example, the same video content will be broadcasted to the entire ONUs but will only used by the requested ONU, introducing an unacceptably high overhead. The proposed architecture uses a proxy on each ONU to buffer the video content so it can be reused by other ONUs instead of being re-requested remotely. This is done though the use of special hardware and modifications to the ONU stack which reduce delays.

The work in [37] solved the special VoD problem by caching the content, and this research shares the same vision. However, a new protocol that takes into consideration downstream broadcast property of the EPON has to be developed to support streaming and interactive applications as well.

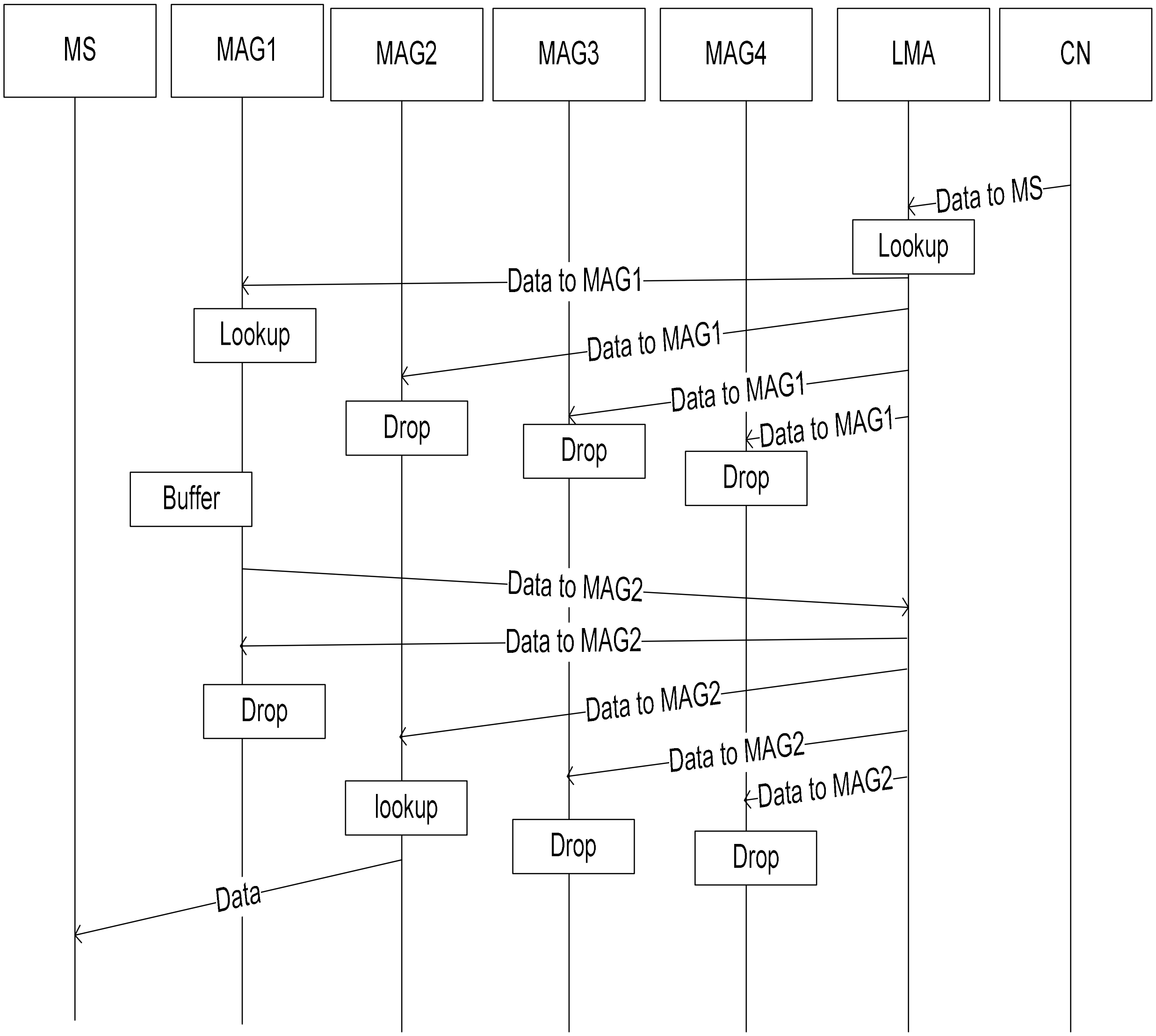


Figure 2.6: A Data Plan of FPMIPv6 with EPON

2.5 Proposed Solution

In this section we will present a novel fast mobile Netlmm that is designed to overcome the problems of combining EPON and PMIP architectures that eliminates packet loss and reduces the data delivery cost.

The ability to assign multiple addresses using virtual ONUs is one to take advantage of: this way, virtual ONUs are treated by the OLT as normal ONUs with different addresses. However, they share the same hardware. This can be used to assign multiple MAGs on the same ONU that is viewed by the OLT as separate entities.

The other aspects of the EPON architecture to be made use of are the broadcasting and multicasting features. EPON provides a Shared Media Emulation (SME) with which a message can be multicast or broadcast to several points by setting a special field in the message frame.

In the proposed architecture, each ONU hosts two MAGs with different roles. The first MAG is responsible for handling all the ptp traffic that is intended for the MNs attached to the MAG. The second MAG only handles the MNs that are in Handoff. This second MAG is invisible to any MNs because the MN communicates with both MAGs via the same Logical Link. The main overall difference between the first and second MAGs is that the second MAG listens on both the multicast address and the node address.

Whenever an MN requests a fast handoff initiation, the second MAG receives the request and sends a PBU to the LMA with the multicast address (i.e. MULTI_MAG_IP) as the source. The LMA will then reply by a PBA that will be captured by both MAGs, however, only the second MAG will react to it as it has the multicast source address.

There are two benefits derived from using these separate MAGs. First is a separation of responsibilities: only the MNs that requested a fast handoff initiation will be handled by the second MAG reducing the lookup time, the volume of processing power required and the cache memory size. The other benefit is that the second MAG shares a multicast address with all the other ONU's respective second MAGs. This means that whenever data is received by LMA, it will be sent to all second MAGs, eliminating the need for expensive data forwarding from old MAGs to the new MAGs as shown in Figure 2.7.

This proposed solution avoids using caching and forwarding as a result of its use of multicasts and buffering on selected ONUs. The main advantage of using multicasting is found within a special feature of the EPON, where all downstream data is multicast to all the ONUs, so there is no additional data overhead incurred from using multicast.

There are three ways to perform the buffering in this system. The first way, utilizes the traditional fast mobile IP protocols: the old MAG communicates with the new MAG and forwards all the packets to the new one where they are buffered. The second way is to create the buffer for all the MNs in handoff on the LMA and OLT side. . Lastly, the data can be buffered on all the neighboring MAGs.

However, there is a problem with the first mode of buffering (using the traditional fast mobile protocol). If the next MAG is wrongly selected due to signal interference, obstacles, or fast moving nodes, a situation arises where all buffered packets are lost and the handoff process needs to be started again from scratch.

This problem can be overcome by caching data in neighboring MAGs. Since each MAG is connected to an ONU and all of them are connected to the same OLT there is no extra overhead as a result of sending the same data to all ONUs. Each ONU in the system has a multicast group setup, each multicast group only include a limited number of the ONU neighbors. The multicast groups are configured statically at the network planning phase. This reduces the number of MAGs that need to buffer the data. With the ability of the IPv6 protocol to support several multicast address groups, each MAG is able to join multiple multicast groups based on their location and loads.

One extra benefit of this solution to the existing fast mobile IP buffering process is that the earlier MAGs and MNs do not need to know the next MAG's information. Since data is buffered on the adjacent nodes, the system guarantees the delivery of the data.

The second approach to buffering is also flawed: buffering all the data on the OLT or LMA sides would result in huge traffic delays. Since all the data sent towards all the vehicles in the network will be cached, regardless of which MAG they are connected to. This increases the cache size dramatically, running the risk of a cache overflow. Furthermore, sending all the buffered data from an OLT to a new MAG would result in a packet delay for other data targeted towards the normal MNs.

The main drawback of the proposed solution is that all the MAGs that are involved in the multicast group will have to search if the MN has already arrived to the MAG or not. However, providing that the high look up speeds that are in range of millions lookups per second and the nature of the proposed solution of isolating the MNs in handoff from the other, eliminates this problem.

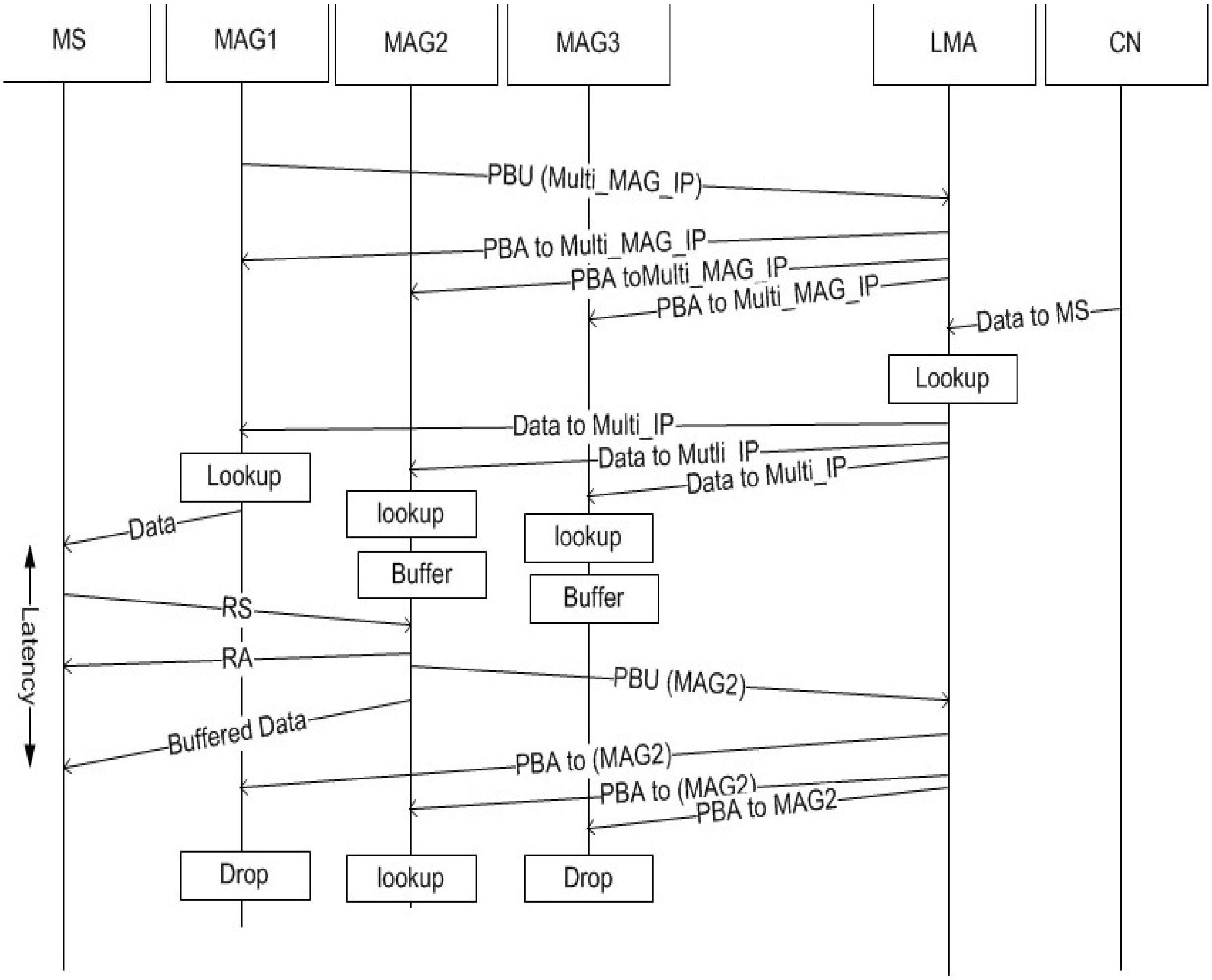


Figure 2.7 : A Data Plan of Proposed Solution

2.6 Mathematical Analysis

In this section we compare the packet delivery overhead of standard, fast mobile PMIPv6 on EPON. This analysis is limited only to the packet delivery overhead, hence we do not take mobility signaling and control signaling into consideration, since they are not key players of EPON's packet overhead. Although, the provided analysis demonstrates the problems with PMIPv6 implementation on EPON, it is valid as well for other data pivoted architecture such as Mobility Access Point (MAP) and routers in hierarchal mobile IPv6 (HMIPv6).

Packet delivery cost represents the overhead added by the system to data between the LMA and the MN. Firstly, the data packet is processed on the LMA and the MN and a lookup is performed to find the MAG that hosts the MN, we donate the LMA processing cost by C_{P_LMA}. PMIPv6 requires a tunnel between the LMA and each MAG which is represented here by C_{tun}. Tunneling costs include encapsulation, de-capsulation, and adding headers to each packet. The transmission cost between the LMA and MAG and between MAG and MN are donated by $C_{T(LMA,MAG)}$ and $C_{T(MAG,MN)}$ respectively.

We define C_{PD} as the total data overhead cost

$$C_{PD} = C_{P\_LMA} + C_{tun} + C_{P\_MAG} + C_{T(LMA,MAG)} + C_{T(MAG,MN)} \quad (1)$$

Let α be the number of packets, tunneling cost is defined in (2) where ξ is the proportional of tunneling header size to the data packet size. $N_{connections}$ are the number of parallel connections, here $N_{connections}$ equals to 1, since we use only one transmission frequency. .

$$C_{tun} = \alpha * (\xi - 1) * C_{T(LMA,MAG)} * N_{connections} \quad (2)$$

where $\xi \geq 1$

We consider C_{P_LMA} implements binary tree search, then the overhead is related to log_2 of the total number of MNs. μ_{lk} is the unit lookup cost.

$$C_{P\_LMA} = \alpha(log_2(N_{MN}))* \mu_{lk} \quad (3)$$

Similarly, processing cost at each MAG C_{P_MAG} is calculated in (4)

$$C_{P\_MAG} = \alpha(log_2(N_{MN_{MAG}}))* \mu_{lk} \quad (4)$$

This research introduces κ_{medium} constant that is related to each media and τ is the unit transmission cost.

$$C_{T(LMA,MAG)} = \alpha(\kappa_{medium} * \tau) \quad (5)$$

For simplicity we will use $C_{T(MAG,LMA)} = C_{T(LMA,MAG)}$

For wireless we equation (5) will become (6)

$$C_{T(MAG,MN)} = \alpha(\kappa_{wl} * \tau) \qquad (6)$$

The total overhead packet delivery cost can be formulated as in (7)

$$C_{Total\,PD}^{PMIP,Standard} = \alpha(log_2(N_{MN})^* \mu_{lk} + log_2(N_{MN_{MAG}})^* \mu_{lk} + \xi * \kappa_{PON} * \tau + \kappa_{wl} * \tau) \qquad (7)$$

In (8) we show the packet delivery cost excluding the wireless portion since it is similar in all the compared protocols.

$$C_{PD}^{PMIP,Standard} = \alpha(log_2(N_{MN})^* \mu_{lk} + log_2(N_{MN_{MAG}})^* \mu_{lk} + \xi * \kappa_{PON} * \tau) \qquad (8)$$

The traditional fast mobile can packet delivery cost can be represented as the normal cost in addition to costs related to forward messages and buffering during the handoff as shown in (9)

$$C_{PD}^{FastMobile} = C_{PD}^{PMIP,Standard} + C_{PD}^{Forward} + C_{PD}^{Buffering} \qquad (9)$$

The packet delivery cost of the forwarding can be calculated as the cost to forward the packet to the LMA and then from LMA to the target MAG as shown in (10) and (11)

$$C_{PD}^{Forward} = C_{T(LMA,MAG)} + C_{PD}^{PMIP,Standard} \qquad (10)$$

$$C_{PD}^{Forward} = \alpha * T_{Handoff}(log_2(N_{MN})^* \mu_{lk} + log_2(N_{MN_{MAG}})^* \mu_{lk} + 2 * \xi * \kappa_{PON} * \tau) \qquad (11)$$

Where $T_{Handoff}$ is the percentage of time the MN will be in handoff. The packet delivery cost for the proposed method can be calculated as the normal cost in addition to costs related to multicast and buffer the data as shown in (12)

$$C_{PD}^{Proposed} = C_{PD}^{PMIP,Standard} + C_{PD}^{Multicast} + C_{PD}^{Buffering} \qquad (12)$$

The Multicast cost can be calculated as the additional MAGs that will have to perform the lookup during the handoff as shown in (13) and (14).

$$C_{PD}^{Multicast} = \alpha((N_{MAG} - 1) * C_{P\_MAG}) \quad (13)$$

$$C_{PD}^{Multicast} = \alpha * T_{Handoff}((N_{MAG} - 1) * log_2(N_{MN_{MAG}})^* \mu_{lk}) \quad (14)$$

In order to find the packet delivery overhead of the fast mobile protocol compared to the proposed, we subtract (12) from (14) $C_{PD}^{FastMobile} - C_{PD}^{Proposed} = C_{PD}^{Forward} - C_{PD}^{Multicast}$ (15)

$C_{PD}^{FastMobile} - C_{PD}^{Proposed} = \alpha * T_{Handoff}(log_2(N_{MN})* \mu_{lk}+log_2(N_{MN_{MAG}})* \mu_{lk} + 2 * \xi * \kappa_{PON} * \tau - (N_{MAG} - 1) * log_2(N_{MN_{MAG}})* \mu_{lk})$ (16)

$C_{PD}^{FastMobile} - C_{PD}^{Proposed} = \alpha * T_{Handoff}(2 * \xi * \kappa_{PON} * \tau + ((2 - N_{MAG}) * log_2(N_{MN_{MAG}}) + log_2(N_{MN})) * \mu_{lk})$ (17)

In (17) there are two main contributors the $\alpha * T_{Handoff}(2 * \xi * \kappa_{PON} * \tau)$ and additional lookup cost. However, the cost to transmit and receive data using PON is much higher than that of the lookup due to the power consumption of the PON and the very high rate of lookup which in the range of hundreds of millions lookups per second for the new routers as shown in Figure 2.8.

In this analysis we used 1.1 for ξ, that is calculated by having an average packet of 500 bytes with additional 50 bytes tunneling, number of MNs to be 100, μ_{lk} to be 1/10,000,000, and number of MAGs set at 32.

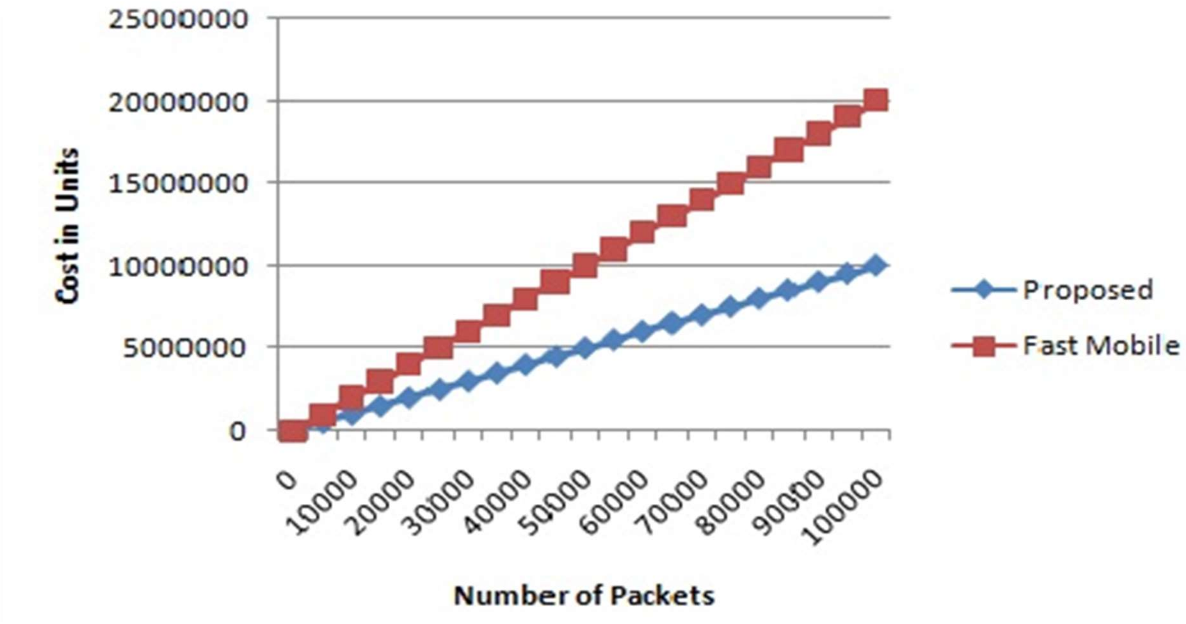


Figure 2.8: Cost of fast mobile versus proposed solution

2.7 **Conclusion**

The integration between EPON and PMIPv6 provides a powerful solution to answer the increasing bandwidth demands of mobile network users. EPON provides the best CAPEX and OPEX in the market, allowing for high bandwidth that can reach 10 Gb/sec. PMIPv6 provides a Netlmm solution that enables seamless mobility to the mobile terminals.

However, the current PMIPv6 protocols do not consider the unique features of the PON. This research identifies the most critical issues and challenges of integrating both EPON with PMIPv6. It shows that fast mobile PMIPv6 does not result in performance enhancements and are not suitable for the current PON architecture because of their unacceptable bandwidth overhead. Evidently, the proposed solution solves the problems of the EPON integration with PMIPv6 without extra data overhead.

2.8 **Future Work**

Further mathematical analysis will be conducted so that the results can be compared under different network traffic scenarios.

The possibility of developing a new dynamic method that groups different MAGs depending on the road traffic, session mobility, and other QoS aspects will also be examined. This investigation will focus on establishing a two layer buffering process where buffering on both OLT and ONUs is performed cooperatively.

CHAPTER 3 ARTICLE 2: ON THE VERIFICATION AND REALIZATION OF VANET APPLICATIONS

Hazem Ahmed
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
hazem.ahmed@polymtl.ca

Samuel Pierre
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
samuel.pierre@polymtl.ca

Alejandro Quintero
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
alejandro.quintero@polymtl.ca

Published in International Journal of Sensors Wireless Communications and Control, 2013.

ABSTRACT

Modeling and realization of Vehicular Ad-Hoc Network (VANET) applications and protocols are very challenging; due to VANET`s unique mobility, driver behavior, signaling and road topology requirements. Although network simulators are commonly used to evaluate wireless network performance, VANET applications require mobility simulators to mimic road traffic topology and mobility. However, network and mobility simulators fall short of evaluating Quality of Experience (QoE) and Quality of Service (QoS) of context-rich applications with multiple user interactions. Using real hardware provides a solid mechanism to evaluate QoE and QoS in real time, though it is very expensive to use in large networks and difficult to reproduce specific VANET scenarios such as collisions and signaling changes due to speed and road topology. Emulation techniques combine the benefits of using both simulation and hardware approaches. Recently, multiple Ad-Hoc emulators are developed, but they are not tailored to VANET`s requirements. In this article, we survey the most suitable network and mobility simulators and their integrations for VANET. Then we compare their features, popularity, challenges and recommend the best ones corresponding to VANET application conditions. Moreover, we perform a comprehensive review on Ad-Hoc emulators showing the benefits and suitability of each one for VANET applications. This article serves as a start point for developing comprehensive VANET emulators and testbeds; it also acts as a start-up guide for researchers to select the best tools for VANET application and protocols modeling, simulation, and emulation.

3.1 **Introduction**

With the broad adoption of new software technologies in vehicles, vehicular applications are poised to be a promising market. VANET applications are classified into two types namely safety and infotainment. Safety and driver assistance applications are critical applications to enhance the road safety and mitigate road hazards, including cooperative collision avoidance, cooperative maneuvering, automated driving at low speed and on highways, and road condition detection [2]. However, the goal of infotainment applications is to enhance the driver and passenger experience by providing information and services such as interactive gaming, audio\video sharing, interactive messaging and news and weather information broadcasting.

There are three types of communications in VANET namely Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and a hybrid between V2V and V2I. In V2V communication, vehicles communicate directly with each other with no infrastructure assistance. This is typically useful for safety applications such as cooperative collision avoidance where vehicles can cooperatively broadcast a safety message to avoid a collision or interactive messaging between passengers on different vehicles. In V2I communication vehicles communicates with roadside units (RSUs) or existing road infrastructure system. V2I applications include Internet service connectivity, road condition reporting, traffic signaling status broadcasting, and road condition warnings.

Network simulators such as NS2 [38] and OMNET++ [39] can be used to evaluate wireless network performance regarding packet loss, messaging delay and jitter. Standalone network simulators can simulate simple mobility patterns such as Manhattan and random waypoint mobility models, however, it fails to simulate complex mobility models such as downtown scenario with intersections and traffic lights. Complex mobility models are achieved using mobility simulators, such as SUMO [40], that simulates driver behavior, road topology, and vehicle maneuvering. Using network simulators along with mobility simulators provide a robust mechanism for evaluating network performance of VANET applications.

Using real hardware for creating a testbed to validate VANET applications can be quite expensive and very hard to obtain. However, it gives the best practical results for the selected test case. For example, testing a VANET interactive game that will be used by two passengers in two different vehicles, will require two-vehicle that are driven with specific speeds in a real environment such as downtown. In this simple example, if it is needed to test the application with different road traffic jams, different speeds, different road and wireless obstacles, and different driver behavior, the experiment cost will increase dramatically.

Emulation is a technique that takes benefits of both real hardware testbeds and simulation. In emulation, an application can run on a testbed in which the underlying communication layer is simulated. For example, by emulating the wireless interface on a mobile node, wireless channel properties such as bandwidth, jitter, and packet loss can be enforced to emulate a specific environment.

In this article, we survey the most suitable network and mobility simulators and their integrations for VANET. Then we compare their features, popularity, challenges and recommend the best ones corresponding to VANET application conditions. Moreover, we perform a comprehensive review on Ad-Hoc emulators showing the benefits and suitability of each one for VANET applications. This article serves as a start point for developing comprehensive VANET emulators and testbeds; it also acts as a start-up guide for researchers to select the best tools for VANET application and protocols modeling, simulation, and emulation.

3.2 **Simulation**

Simulation is the most suitable technique to conduct large-scale experiments for the purpose of protocol evaluation. In this section, firstly we discuss the available network simulators and analyze their suitability for VANET applications. Then, we demonstrate common mobility simulators and investigate their advantages and disadvantages. Finally, we study the integration platforms between network and mobility simulators.

3.2.1 Network Simulators

Network simulation is very useful in evaluating wireless signaling and channel QoS. Network simulators can simulate large networks and special cases that are not easily reproducible on the real world. Moreover, simulation scenarios are very easy to modify which make simulation a robust and cost effective way of testing application performance.

In this section, we selected NS2, NS3, GloMoSim [41] and QualNet [42], and J-sim [43], [44] to study. Although, there are other accurate simulators in the market, the selected ones are the most common and can be used for free for educational purposes.

A NS2 and NS3

NS2 is a discrete event simulation that is widely used for network simulation. It is on of dominant network simulator that is widely accepted by the academic research community. Modules in NS2 are developed in C++ and TCL scripting language. There is a large number of libraries available for NS2 that is contributed by researchers.

The NS2 simulation modules and their results are highly accurate due to a large number of performance evaluations conducted on them. However, NS2 lacks the clear architecture and modularity, and it is complex to understand and develop new protocols. Moreover, it suffers from changes and patches done from a version to another, for example, a module that is contributed and patched to a specific version may be not as easy to be used is a different one. NS2 provides modules for 802.11p simulation as well as limited support for LTE simulation [45], which are essential for VANET applications. NS2 is retired on 2011, and all efforts are concentrated on NS3

NS3 is not just a newer version of NS2, but it is a completely revamped simulator that is built with a clear object-oriented layered architecture. Although not all modules developed for NS2 are converted to NS3, currently there is a rich library of modules supported in NS3. NS3 is using C++ with Python for the development of modules. Multiple performance analysis experiments are conducted to compare NS2 to NS3 show the superiority of NS3 in means of accuracy, processing time and memory management.

NS3 provides excellent support for LTE, urban mobility and obstacles modeling which are essential for complex VANET simulation. Using parallel and distributed processing NS3 can reach 360 million nodes [46] in a simulation, this makes it suitable for very large network simulation.

B OMNET++

OMNET++ provides a modular and easy to use simulation environment. It consists of hierarchically nested modules that are connected using a high-level scripting language files (.NED) that are easy to understand. OMNET++ provides two useful modules namely: INET for wireless simulation and Mobility Framework for MANET. Furthermore, It provides 802.11p modules and limited support for LTE simulation using the simuLTE module [39]. There is sufficient documentation available of how to use OMNET++ with Eclipse IDE for rapid development of simulations. OMNET++ is used to simulate small to large networks since it supports parallel simulation. Besides, OMNET++ comes with easy to use debugging and visualization tools.

C GloMoSim and QualNet

Global Mobile Information System Simulator (GloMoSim) is a network simulator that is famous for wireless network simulation. GloMoSim is widely known for simulating huge networks (millions of nodes). This is due to its ability of parallel network simulation. GloMoSim is known to be suitable for wireless simulation more than wired one.

GloMoSim is programmed using a language called PARallel Simulation Environment for Complex Systems (PARSEC), which is a C-based language for parallel and sequential event simulation, developed by the Parallel Computing Laboratory at UCLA. The development of GloMoSim stopped in 2000, and the amount of available documentation is limited. GloMoSim does not provide neither LTE nor 802.11p modules; that limits its ability to simulate complex VANETs nor IVC. By using external tools such as obstacle mobility model project, a Voronoi graphs can be designed to represent obstacles [47].

QualNet is the commercial version of GloMoSim and was developed after GloMoSim stopped in 2000. It represents a layered simulation environment with clear documentation and modules. Qualnet supports geographical routing algorithms such as location aided routing and landmark routing. Also, it provides a rich library and models for simulating urban wireless scenarios including obstacles and urban wireless fading modeling. QualNet also supports the LTE simulation. However, there is no module for 802.11p protocol.

D J-Sim

J-Sim[44] is a Java-based network simulator (formerly called JavaSim). J-Sim is built based on the component based architecture, where there is a standard interface to integrate modular simulation components. Similar to NS2, J-Sim uses TCL to build components; however, instead of the C++, J-Sim uses Java as a programming language to implement the component logic. The key advantage of J-Sim over NS2 are its modularity and ease of use.

In a performance comparison of J-Sim against NS2, J-Sim shows better memory usage and better execution time when running an experiment with GPRS protocol [43], though, unlike there is no extensive performance study about the accuracy and performance of J-Sim.

One of the main disadvantage of J-Sim compared to other network simulators, is the very limited number of available modules and protocols built on the simulator. Moreover, there are only two mobility models provided with J-Sim namely Random Waypoint and Trajectory Based mobility models and no support for external mobility generators.

J-Sim is not suitable for complex VANET simulation requirements since it does not support integration with other urban traffic simulators, it only supports a couple of wireless routing protocols, it does not support neither LTE nor 802.11p, and finally its development has stopped. J-Sim can be used to simulate VANET to generate quick simulations with random waypoint mobility.

E Analysis and Recommendation

In this section, we compare the popularity of selected simulators in the research domain. Then we conduct a feature comparison between each one of them stressing on the VANET application requirements. Finally, we present the recommendations.

I Popularity

We conducted a search for all publications in the VANET domain that used OMNET++, J-sim, NS3, and NS2 to conduct performance analysis between 2009 to date. We considered popular research libraries such as (IEEE, Wiley, Springer, ACM, and Elsevier).

As shown in Fig. 3.1, NS2 is the most popular simulator; due to its accuracy and the availability of contributed modules that that can be reused to conduct performance comparison or modified to a new algorithm. Also, NS3 is growing in the second place due to its modularity, standardization and the quality of simulation. Moreover, for new protocols, NS3 is a better choice compared to NS2 due to its simplicity and future development.

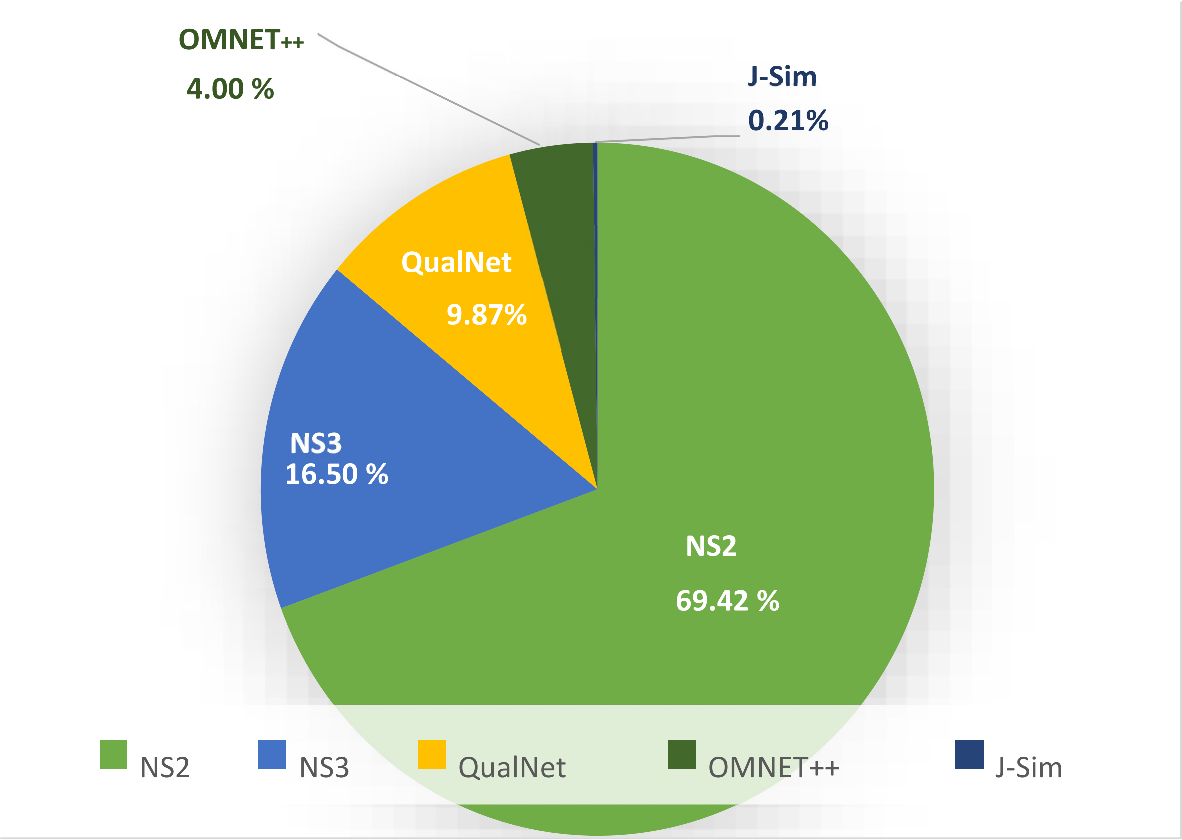


Figure 3.1: Network Simulator Popularity

In Fig. 3.2, we rank the probability we show the popularity of each simulator by year. NS3 popularity is growing year after year while NS2 is decreasing. Since new protocols are implemented in NS3 and older modules are being converted to NS3. OMNET++ is also growing slowly as compared to NS3.

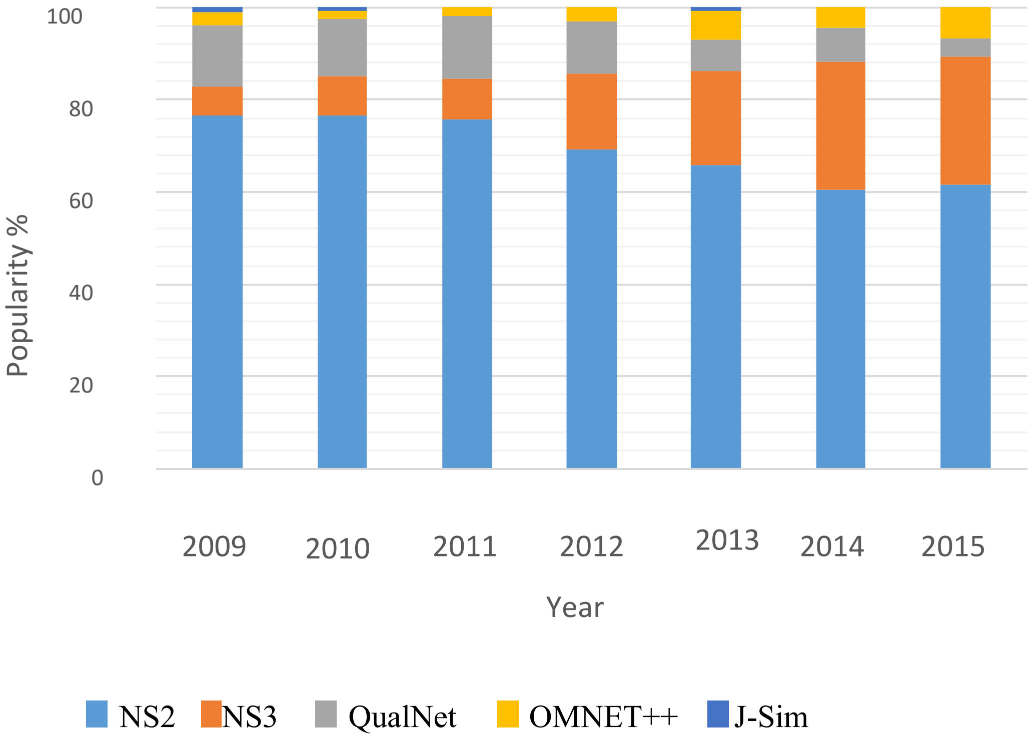


Figure 3.2: Network Simulator Popularity by Year

II Feature Comparison

Network simulators have several differences in their architecture, structure, accuracy, and performance. As shown in Table 3.1, we selected general comparison features such as easiness, language, support, open source status, and current status of the simulator. Easiness is a relative matter that can change from one to another based on experience and background. Though well organized, modular, and object-oriented platforms are considered easier to use. On one hand, NS2 is seen as harder than NS3 since it is mixing OTCL with both C and C++ and it is not as straight forward as NS3. On the other hand, J-Sim is considered the easiest due to its simplicity and Java usage. The support feature is related to the availability of extensive documentation, active discussion and support groups and the availability of tutorials and examples.

Table 3.1: Network Simulator Feature Comparison

	NS2	**NS3**	**OMNET++**	**QualNet**	**GloMoSim**	**J-Sim**
Easiness	Hard	Moderate	Easy	Easy	Moderate	Easiest
Language	C++\ oTCL	C++\ Python	C++	Parsc (C)	Parsc (C)	Java
Scaling	Large (<8000 nodes)	Very large	Large	Very large Millions of nodes	Very large	Small
Support	Excellent	Excellent	Good	Excellent	poor	Poor
Geo-routing support	Best	Very good	Good	Very Good	No	No
Obstacle &Urban Mobility	Good	Very good	Poor	Best	Good	Very limited
LTE support	Limited	Very good	Good	Very good	No	No
802.11 p	Yes	Yes	Yes	No	No	No
Open Source	Yes	Yes	Yes	No	Yes	No
Commercial	No	No	No	Yes	No	No
Status	Retired (2011)	Active	Active	Active	Stopped (2000)	Stopped (2006)

The VANET specific features selected are the availability of geo-routing protocols, obstacle and urban mobility modeling, LTE support, and 802.11p protocol support. Geo-routing protocols can be used to route messages in V2V environments. Obstacle and mobility support is an important simulation aspect for VANET. Obstacles can change radio propagation and affect the signal strength between communicating vehicles.

The Wireless Access in Vehicular Environments (WAVE)or 802.11p standard, is an important standard for VANETS. NS3 and QualNet provide excellent support for 802.11p. However, OMNET++ and NS2 provide less support. Finally, LTE is poised to be the future for V2I applications. Hence, it is an important feature for choosing a network simulator.

III Recommendations

NS3 is poised to be the best simulator to support VANETs due to its modularity, ease of development and the research community adoption. If a new module or protocol needs to be developed, we recommend using NS3. However, we have found out that there are many Ad-Hoc routing protocols t published on previous research, and most of them are implemented in NS2 and not NS3. If there is a need for a performance analysis of these protocols, then NS2 should be a suitable choice. Another alternative is OMNET++ that is very easy to use and can generate results quickly, however, it is not as widely accepted compared to NS2 and NS3.

We recommend GloMoSim only for simulations of very large networks to generate quick simulations. QualNet provides excellent support for urban mobility modeling, which makes it a strong alternative for NS3. However, since it is commercial, it is not as widely adopted as NS3.

3.2.2 Mobility Simulators

A SUMO

Simulation of urban mobility (SUMO) [40] is an open source road traffic simulator. It is developed to simulate microscopic traffic with several driver behavior, multi-lane, traffic lights, speeds, different vehicle types, accident-free simulation, and much more sophisticated road traffic simulation parameters. SUMO can handle large road traffic simulation with thousands of roads and vehicles.

Realistic maps can be generated for SUMO using tools shipped with the simulator.For example, it supports the open source Topologically Integrated GEographic Encoding and Referencing (Tiger) and OpenStreet maps, in addition to simple maps such as Manhattan like girds, spider shape grids, and random maps.

SUMO is built using C++, and all the control is configurations and controls are done through XML configuration files. It provides tools that capable of converting output mobility traces to be consumed by network simulators. The TraceExporter tool is capable of converting the SUMO mobility traces to NS2, NS3 and OMNET++ mobility trace formats. This is a huge advantage since it enriches traditional network simulators with realistic and complex vehicular scenarios.

For complex road traffic simulation, SUMO provides a Traffic Control Interface (TraCI) plugin. This plugin enables the user to get information about each element of the simulation such as the location of vehicles, speeds, and traffic light state. Also, it allows changing the mobility simulation parameters. This can be very effective coupling SUMO with a network simulator. For example, a vehicle can be commanded to make a lane change or change its speed based on an external command.

B VANETMobiSim

Similar to SUMO, VANETMobiSim [48] provides microscopic rich road traffic simulation. It is an open source project which is developed using Java; this makes it very portable and easy to modify. It supports mobility trace generation for GloMoSim, NS2, and QualNet. Although, it can parse real maps such as Tiger and use them in simulation and its generated mobility traces are very realistic, the tools provided for VANETMobiSim are not as extensive as SUMO. For example, it does not have a tool like “TraCI” where the user (or a network simulator) can control and change the behavior of the simulation.

C VISSIM

VISSIM[49] is a commercial traffic simulator that provides feature-rich traffic simulation. It is capable of rendering 3D traffic scenarios for cars, buses and pedestrian traffic. Similar to SUMO VISSIM can use real traffic maps, however, it is not an open source project so modules can only be controlled using provided configuration parameters.

VISSIM provides rich capabilities for modeling driver behavior and interaction with external applications. It uses COM interfaces to retrieve elements in the simulation such as vehicles, speeds, traffic light status, and all road topology parameters. They allow external application to control the simulation and add external driver and emission models.

D Analysis and Recommendation

I Popularity

We conducted a search for all publications in the VANET domain that are using real traffic simulation tools. The search cover all scientific papers published from 2009 to now as shown in Fig. 3.3. The search indicates that SUMO is the most popular traffic simulator with 57% of all papers used. This is mainly due to its powerful tools of integration with network simulators, being open source, and ease of use. Then VANETMobiSim with 26% on the second place for the reasons mentioned above. Finally, VISSIM with 17% which is a good percentage, taking into consideration it is a commercial simulator.

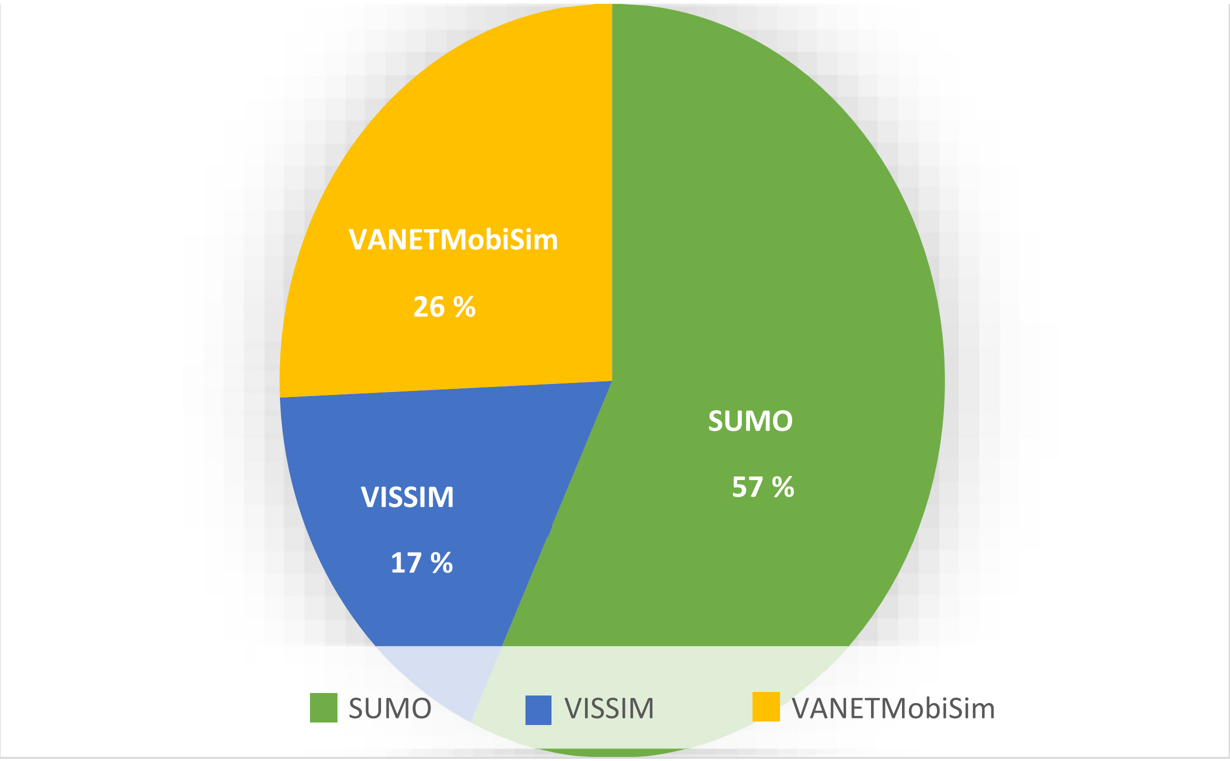


Figure 3.3: Mobility Simulator Popularity

Fig. 3.4 shows the mobility simulator popularity by year. As shown, SUMO is becoming more popular year after year at the expense of VanetMobiSim.

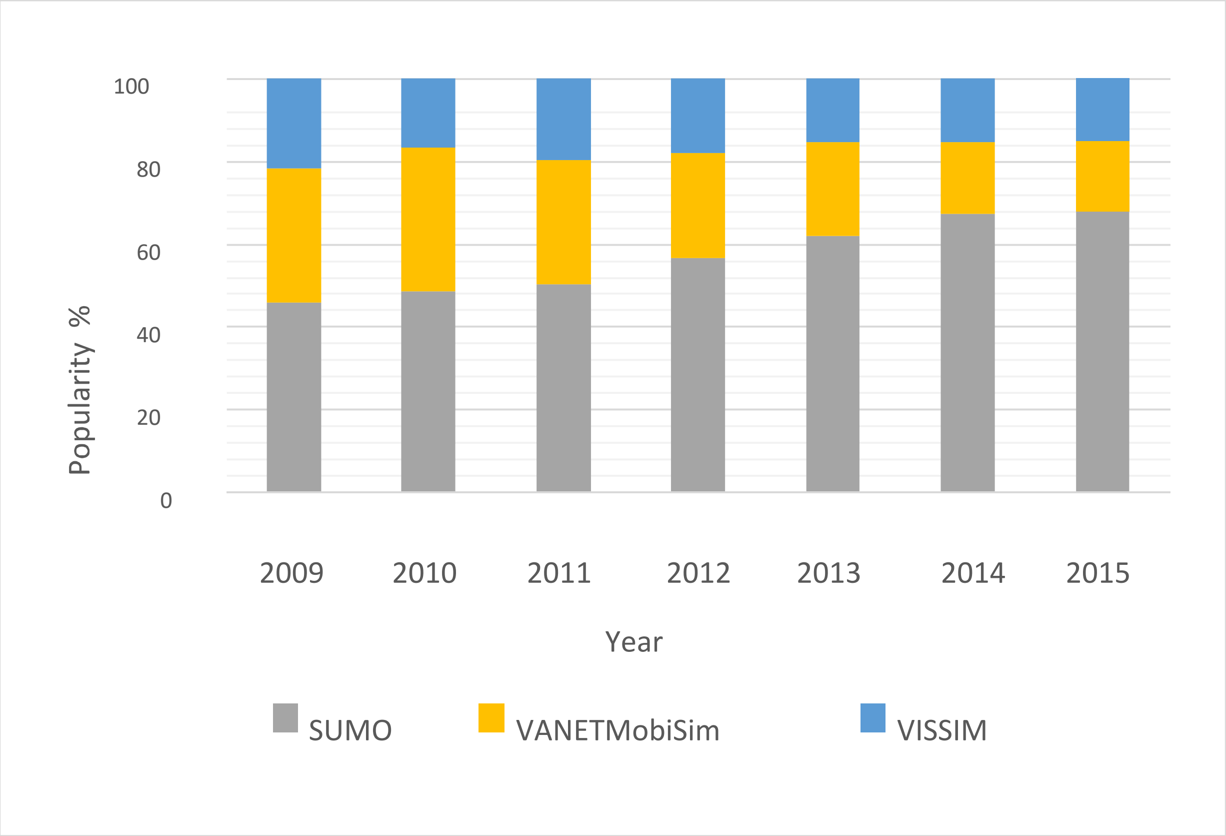


Figure 3.4: Mobility Simulator Popularity by Year

II Feature Comparison

Mobility simulators differ from each other based on several features including their architecture and how they handle macroscopic and microscopic mobility. As shown in Table 3.2, we select comparison features that are more suitable for a basic VANET simulation. In addition to general features such as commercial and open source statutes and ease of use, we use mobility specific features namely: support for real maps, mobility output traces, overtaking criteria, and collision free mobility. Support for real maps is an important characteristic to ensure the practicality of a simulation instead of basing simulation only on unrealistic mobility models.

Output mobility traces and their suitability for different network simulators are vital for the integration with network simulators done by a tool shipped with the mobility simulator or publically available. It worth to mention that there is research done for integration VISSIM with network simulators where they converted the output to be used by NS2 and QualNet, however, these tools are not shipped with VISSIM and not available online.

Collision free and overtaking criteria mobility are two important features for supporting VANET safety applications, such as crash collision avoidance and lane change simulation. Collision free mobility is useful if it is needed to ignore any collision between vehicles in a simulation.

Table 3.2: Mobility Simulator Feature Comparison

	SUMO	**VANETMobiSim**	**VISSIM**
Open source	Yes	Yes	No
Commercial	No	No	Yes
Ease of Use	Moderate	Easy	Easy
Support for Real Maps	Yes	Yes	Yes
Output Mobility Traces	NS2, NS3, OMNET++, QualNet, XML	NS2, GloMoSim, QualNet, XML	__
Overtaking Criteria	No	Yes	Yes
Collision Free Mobility	Yes	No	No

III Recommendations

For wireless network simulations with complex mobility, we recommend SUMO. This is due to its popularity, being open source, ease of use, integration tools to network simulators, and finally, the ability to create complex simulations with two-way communication between SUMO and network simulation can be established. VANETMobiSim is very similar to SUMO in being open source and support integration with network simulators, our experience with VANETMobiSim shows that it is slightly easier to use compared to SUMO. However, unlike SUMO, it does not support tools for two-way communication with a network simulator. So it is recommended in case of the need for mobility traces to be fed later to a network simulator. VISSIM provides extensive tools and protocols for microscopic traffic management, another real benefit of using VISSIM is that it enables users of building external models for driver and emission and support external components to communicate via COM interfaces. However, being a commercial tool limits its comparative advantage as a VANET simulator.

3.2.3 Integration Between Network and Mobility Simulators

With the emerge of VANET applications, there is a need for integration between network simulators and road traffic mobility simulators. There are three categories of integrations namely: One way, bidirectional, and embedded. One way integration is that mobility traces are generated from a mobility simulator and then fed to a network simulator. There is no interaction between network and mobility simulator. This is suitable for network-centric simulations where only the mobility traces are required to verify new wireless applications and protocols. Bi-directional integration is the ability to send commands from network to mobility simulator and vice-versa. However, embedded integration platforms are the ability of a network simulator to contain fully a mobility simulator with no tools needed.

A TraNS

Traffic and Network Simulation Environment (TraNS) [50], is an integration tool between SUMO simulator and NS2. TraNS supports two modes of operations network-centric mode where mobility traces are extracted from SUMO first and then are fed to NS2 [50]. The second mode is application-centric where the mobility can change based on wireless simulator events [50].

For the majority of VANET applications, the network-centric mode will be suitable. However, for safety application that requires changing the mobility behavior when it receives a safety, then application-centric should be used. TraNS provides an easy to use GUI that is helpful in configuring SUMO and NS2. However, TraNS is not compatible with all versions of NS2 and SUMO. Hence, it is critical to use only compatible versions.

B Veins

The vehicle in Network Simulator (Veins) [51], provides integration between OMNET++ and SUMO. Veins is an advanced simulator that supports bi-directional communication between network and mobility simulators. Since OMNET++ is an event based network simulator, it is much easier to integrate with external mobility simulators.

It is evident that if OMNET++ is selected as a wireless network simulator, Veins is essential for generating an efficient simulation. Moreover, if the user has no preference to which network simulator to use, we recommend using Veins with OMNET++ due to its simplicity.

C iTERIS

Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (iTERIS) [52] integrates both NS3 and SUMO. The main benefit of iTERIS over Veins and TraNS is its modularity and that the integration entity between NS3 and SUMO is decoupled from both of them [52].

iTERIS supports bidirectional communication between mobility and wireless simulators, it is available for free download after filling a request form, and it is an open source as well [52]. If the VANET application requires bidirectional communication, we strongly recommend iTERIS with NS3, however, for simple VANET simulations with NS3, mobility traces can be generated by SUMO first and then fed to NS3.

3.3 Emulation

Although simulation is a very efficient and economical technique to validate extensive simulations and hard to reproduce scenarios such as road accidents, it suffers from artificial models that are not always imitating the real environment. Also, some programs must run on the actual machine; this requires combining both real machines with large simulation.

There are three approaches for emulation namely: hardware, a simulator with emulation extension and software-based simulators. In this section, we will describe each one of them, focusing on their suitability for VANET applications. Finally, we compare the features of these emulators to decide the most suitable emulator for the VANET application under test.

3.3.1 Hardware/Real Emulator

Hardware emulator requires real devices to run an experiment. The main benefits of using hardware are the avoidance of any overhead introduced by simulation tools. Moreover, they run the application to be tested natively and do not need special calibration such as clock synchronization. The main drawbacks of using hardware emulators are cost, scalability, and emulation of complex environments.

The cost of using hardware emulation is of course very expensive. For example, if we need to emulate a new protocol for V2V application, using hardware only approach needed to setup two cars that are driven on track with a specified configuration, speed, distance, driver behavior, obstacles and all other parameters that can affect the communication between the two vehicles.

Scalability of the experiment is very limited; imagine we conduct the same experiment on ten vehicles instead of only two in the previous example, what if we increased this number to 100 vehicles to cover a small portion of a metropolitan downtown? .Hence, It is clear that scalability is an issue.

Depending on the application under test, real world emulation could be dangerous or even not possible due to so many parameters that have to be controlled. For example, to emulate a safety application that requires vehicle collision on several road traffic topologies, several driver behavior, several road conditions, and several communication media is not possible using hardware only approach.

3.3.2 Simulators with Emulation Extension

Network simulators such as NS-2, NS-3, and OMNET++ provide emulation extension to enable configuring multiple nodes on the real network that are fully controlled by the simulator. This approach is solving all the problems mentioned for the hardware approach since network simulators can be configured to support thousands of network nodes and can be integrated with mobility vehicle simulators (e.g. SUMO) to support complex road traffic and driver behavior experiments.

There are differences in the emulation approach for each simulator. For example, NS-3 is using the same packets and data used in real networks as compared to NS-2 or OMNET++. This enhances the emulation to simulation integration since there is no need to translate packets from and to real networks.

However, this approach is facing many challenges:

a) It does not represent exactly the real environment since simulation processing has to be executed on the simulation machine to communicate with real networks. This makes testing real-time applications not realistic.

b) It cannot be used to evaluate high-level user applications on all nodes since most network simulators are concerned with communication protocols.

c) It is not possible to assess and use the application with real users unless emulated machines are used.

d) It does not have the ability to emulate context-aware applications.

e) Synchronization between network nodes is very challenging since most network simulators are using virtual clocks.

3.3.3 Software Emulators

These emulators are used as modules or add-on applications to current hardware or virtual systems. The main benefit of this approach is that it works transparently with the application under test. For example, link emulators such as NetShaper [53] can be used to enforce delay, bandwidth, and jitter to a specific nodes or links.

This enables the emulation of context-aware applications, where an application can run on a real hardware that consumes data from multiple sensors that could be virtualized or not and enforcing specific network parameters to emulate real life signaling experience.

Moreover, using real devices (e.g. smart phones) to test the application in an emulation environment enables the user to evaluate the Quality of Experience (QoE) of the application.

However, using this approach is facing some challenges:

a) It is not easy to control all the emulation parameters on all nodes.

b) Overhead introduced by specific protocol stack emulation could influence the performance.

c) The complex context-aware application requires a significant amount of sensor data that are related to each environment.

d) Synchronization between real hardware and virtual emulated nodes could be a challenge for real-time applications.

3.3.4 Summary and Discussion

In this section, we compare the three emulation approaches as shown in Table 3.3. Context-aware application emulation is an important feature to test the integration among different sensors and events. Using hardware emulators, it is very hard to achieve all sensors to cooperate. However, for simulators with emulation extensions, only the emulated devices can be utilized as context-aware devices under test. Software emulation provides the best fit for context-aware VANET applications since all sensors can be emulated interactively.

Table 3.3: Emulation Type Comparison

Approach / **Criteria**	**Hardware Emulation**	**Simulators with emulation extension**	**Software Emulation**
Context-aware emulation	Hard	Limited	Easy
Scalability	No	Yes	Yes
Cost	Expensive	Cheap	Moderate
Accuracy	Best	Good	Better
High level Application test	Yes	Limited	Yes
Simulation overhead	No	High	Low
Mobility Support	Very Hard	Moderate	Moderate/Easy
Complex scenarios	Very Hard	Easy	Moderate

Scalability feature is defined as the ability to support large networks, and the cost is how expensive to run several experiments. Accuracy in hardware emulators for an experiment that imitates exactly the real life scenario is the best and software emulation accuracy is better than simulators with emulation extension due to the simulator overhead. High-level application test is the ability to test VANET applications as is. This is the easily achievable with software emulators, and hardware emulators but limited only to emulated terminals in the simulators with emulation extension case. Complex environment scenario and mobility support are much easier with simulators with emulation extension and very hard with hardware emulators.

3.3.5 Quality of Service and P2P Link Emulation

This section will study some well-known case studies of the implementation and research of successful network emulations. Note that this article places emphasis on the software emulation approach since it is more realistic in to be used as a cornerstone for a V2V testbed.

The Open Access Research Testbed for Next-Generation Wireless Networks (ORBIT)[54] is a hardware-based network emulation testbed. It consists of 802.11 wireless nodes that are arranged in a two-dimensional grid. Each node contains wireless interfaces and Ethernet ports. For emulation purposes, a framework is developed to enable users to define a scenario, then deploy the scenario on nodes, run experiments and finally obtain the results. ORBIT is very useful for conducting analyses of new protocols; however, it lacks the ability to emulate complex mobility and context-aware applications. Also, the test environment is not controllable, since it is based on real hardware interfaces.

PlanetLab [55] consists of a set of virtual machines that are scattered and deployed globally. These virtual machines are divided into smaller slices geographically distributed where they can be accessed to test applications globally. PlanetLab provides a useful resource for testing internet-enabled applications. However, it is originally designed to handle wired networks, and it is not suitable to handle wireless ones.

DummyNet [56] is a simple, lightweight and efficient traffic shaping tool. It works as an intercepting filter to add delay or jitter and to limit bandwidth. The implementation consists of less than 300 lines of code and is available under a FreeBSD license. DummyNet is one of the first attempts to bridge the gap between simulation and emulation by providing traffic-shaping functions seamlessly to an application under test.

NetShaper [53] is similar to Dummynet in that it emulates link layer properties, but it is also developed as a loadable kernel module. The main difference between NetShaper and DummyNet is NetShaper's support to dynamic network configuration. NetShaper contains a configuration utility application that runs on the user machine to control the link layer emulation's parameters. It communicates with control nodes to receive control commands using UDP messages. The control nodes are configured to change the network configuration in a dynamic manner.

Simple Emulation of Delay and Loss for Ad Hoc Networks Environment (SEDLANE) [23] is an extension for DummyNet in which NS-2 simulations are used as a source for delay and packet loss information that DummyNet requires operating. The idea is to use NS-2 data to simulate complex scenarios that would be tough, if not impossible, to replicate in real environments. Using network simulator outputs as an input for a link emulator allows for the simulation of complex traffic scenarios, obstacles and signal properties in different settings.

KauNet [24] can be considered an extension for DummyNet. It provides the ability to create traffic shaping profiles that can be data-driven and discrete data-driven for DummyNet. These configuration files ensure the reproducibility of the traffic shaping experience. Moreover, KauNet provides the ability to introduce bit errors, which is not supported by DummyNet. There is a slight performance difference between KauNet and DummyNet, an experiment performance analysis of which is studied in [25].

NISTNet[26] provides a robust traffic shaping tool. It is available in Linux kernel 2.4 and is patched to higher kernel versions. Similar to DummyNet and KauNet, NISTNet provides tools to throttle bandwidth, add packet delay and apply jitter. Moreover, it provides a GUI to configure traffic shaping parameters.

NetEM[27] and traffic control (TC) tools are widely used and integrated into Linux kernels. They are very similar in concept to NISTNet. However, NetEM and TC are superior in terms of the number of operating systems that support them. For example, Android framework currently supporting the TC and NetEM commands, along with most Linux distributions (with patching required).For an experiment-based performance analysis comparing DummyNet, NetEM, and NISTNet, refer to [28].

Conchon, E., et al. proposed W-NINE[29]: a two-stage emulation platform for mobile and wireless systems. Simulation for a preconfigured network with the nodes and their mobility can be conducted using Simulator for Wireless Networks Emulation (SWINE) to predict jitter, bandwidth, and other network characteristics. The output from the first stage is then used as input for a KauNet [24] controller router that applies these effects upon the nodes participating in an emulation. The idea of using two-stage simulation is time-consuming but very effective in reducing the time that a simulator needs to process the information. However, SWINE has not been updated, and its suitability for supporting and integrating with other road traffic mobility's simulators is not clear.

The idea of using multistage emulation is praiseworthy, but overall, a more generic platform that can integrate with different traffic and mobility simulators and utilize captured real life wireless traces.

A Summary and Discussion

In Table 3.4, we compare the features of the selection link emulators. The Environment feature is vital in determining what the supported platforms are. For example, NetEM is supported by Android systems. Hence, it will be much easier to use it to emulate Android systems. Whether hardware is required in the emulator is another feature that may limit the scenarios that can be emulated. For VANET, mobility is the most important aspect of emulation. Emulab is using real robots that will move in a controlled environment. However, for software emulator such as SEDLANE, the mobility traces can be used to generate wireless traces to emulate mobility.

Table 3.4: P2P Emulator Comparison

Name	Environment	Hardware Required	Mobility	Emulation Type
Mobile EmuLab	802.11	Yes	Robots	Real
ORBIT	802.11	Yes	No	Real
DummyNet	IP layer, FreeBSD	No	No	Traffic Shaper/ Manual Input
KauNet	FreeBSD	No	No	DummyNet/Configuration scripts
SEDLANE	FreeBSD	No	Possible	Central link emulation
W -NINE	Linux	No	Possible	KauNet, SWINE Traces
NetShaper	Linux	No	No	Traffic Shaper
NetEm/TC	Linux, Android	No	No	Traffic Shaper
NIST Net	FreeBSD, Linux	No	No	Traffic Shaper

3.3.6 Challenges of using Software Emulation

A Obtaining network traces

Obtaining traces is an important factor to ensure that a network emulator is performing as realistic as possible. There are mainly three main categories of network traces namely, real traces, simulator output traces, and synthetic traces.

Real network traces are the traces captured from the actual network by using a logging agent or a man on the middle technique. The benefits of using them are to run the experiment on real environment. These real traces can be employed as background traffic[57] or used to reply specific scenarios. However, the real traces are lacking the reconfigurability and the agility of defining special case scenarios and large-scale examples. One of the most intensive network traces available online is Community Resource for Archiving Wireless Data at Dartmouth (CrawDad) [58].

Network simulators such as NS2 and NS3 has are very efficient in creating wireless traces. These traces can be used as inputs for an emulator. However, these traces are not a good representation of real world scenario and only can be useful in emulating general cases with large network scenarios.

Using mobility simulators along with real world topology and traffic information results in realistic synthetic simulations [59]. In [59], it is proven that using SUMO with real traffic information collected by road sensors can generate synthetic data that takes the advantage of both real world scenarios and the efficiency of network simulators.

B Filtering synthetic and real traces scenarios

Due to the significant amount of traces and information in real network traces and/or synthetic information produced by network simulators, it is a challenge to extract the desired experiment scenario. This mainly affects the repeatability of an experiment and reduce its realistic value. For example, if we had real network traces that are captured day long, these traces will contain different load criteria such as collision, bottlenecks, special cases and network overload at a specific point in time.

In [60] the authors address the problem of filtering the large traces available for specific simulation profile. This is done by using a set of classifiers that can filter a large amount of traces available using data stochastic clustering and deviation analysis. For example, a researcher may be interested in highly loaded traffic traces; these classifiers can select such scenarios from different traces.

In [61] the authors use a model checker to discard ns-2 run scenarios that do not meet the simulation requirements provided a designer. Using an XML file the designer provide the objective parameters. First, all the possible ns-2 runs are generated, and then the model checker will filter these scenarios, and only the scenarios that confirm to the designer objective are run. This approach reduces the time of simulation and focuses on the important simulation analysis.

In [62] the authors use actual wireless traces with a formal verification engine to limit traffic simulator (NS2) from generating synthetic simulation results that are far from reality compared to the real traces. The authors use the verification engine to stop network simulator from going beyond the actual traces range.

3.4 Conclusion

Modeling and verification are important factors in the evolvement of VANET application development and adoption. In this article, we studied the most common VANET wireless and mobility simulators and their integration platforms extensively. Moreover, we demonstrated different emulation techniques for VANET and studied their benefits and limitations thoroughly. This article serves as a start point for developing comprehensive VANET testbeds; it also acts as a start-up guide for researchers to select the best tools for VANET application and protocols modeling, simulation, and emulation.

CHAPTER 4 ARTICLE 3: A FLEXIBLE TESTBED ARCHITECTURE FOR VANET

Hazem Ahmed
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
hazem.ahmed@polymtl.ca

Samuel Pierre
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
samuel.pierre@polymtl.ca

Alejandro Quintero
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
alejandro.quintero@polymtl.ca

Published in Journal: Vehicular Communications, Elsevier, 2017.

ABSTRACT

Verification of VANET applications and protocols is a challenging task due to its unique mobility, driver behavior and networking requirements. Although network simulators are widely used for network performance evaluation, there is a lack of a realistic testbed that is capable of emulating the VANET environment, enabling user testing and evaluating both Quality of Service (QoS) and

Quality of Experience (QoE). This article studies the requirements of such a testbed and introduces a flexible VANET testbed architecture that is tailored for VANET applications' needs. The implementation of this architecture is tested using standard VANET applications to evaluate its feasibility for vehicular applications. Our results confirm the suitability of the proposed testbed to meet VANET requirements. Furthermore, the effect of adding a caching entity is experimented and the results show its ability to mitigate the testbed's overhead.

4.1 Introduction

The world has more than 1 billion active Android users as of June, 2014[63], with more than one million devices being activated every day[63]. With the mass production of such smart technology (typically in the form of phones and tablets), the cost to manufacture these platforms is constantly getting cheaper even as devices become more powerful and capable.

With this dramatic growth in the number of portable wireless devices, there is a huge market for the development of smart applications that can interact with users' lifestyles and provide context-aware interactions. Mobile games markets are growing as well; it is expected to reach $23 billion by 2016[64].

In the last five years, automakers adopted integration platforms between the onboard devices and personal smartphones. Hence, the line between consumer electronic devices (e.g. smartphones) and onboard vehicles has become thinner. Furthermore, multiple applications are targeting consumer electronics for VANET applications.

The integration between onboard devices and consumer electronics offers more flexibility for applications that can be used on a vehicle and get best user experience of applications on the vehicle. Smart Device Link (SDL)[65] is one popular example that is adapted by Ford and Toyota. SDL enables running applications on smartphones on the vehicle's head unit. Other examples are Google's Android Auto and Apple's CarPlay that allow using the car's head unit to display and control applications on the consumer devices. Also, developers can develop VANET applications are that are suitable for Android Auto or Carplay.

Testing and validation of context-aware and mobile application is a huge market. ABI research firm mentioned that mobile testing tools generated $200 million of revenues in 2012. With the new mobile automated testing tools, this number will reach $800 million by the end of 2017[66].

Originally network simulators such as NS-2 and OMNET++ were enhanced by supporting both wireless interactions and basic mobility for simulated nodes. Although these simple mobility protocols work well for simple Mobile Ad-hoc Networks (MANETs), it failed to handle the VANET mobility requirements. VANET requires complex mobility models involves road topology, driver behavior and different traffic conditions and routes. This leads to the integration between network simulators and urban mobility simulators such as SUMO which is capable of simulating complex traffic scenarios and sophisticated driver behavior. This integration enriches the VANET simulation by using realistic road topology and mobility conditions.

Although the ability of network simulators integration with urban simulators to simulate complex VANET scenarios that are very difficult, if not impossible, to replicate in real environments, it fails to simulate complex user interactions, software application response, context-aware applications and on device testing. Furthermore, simulators can result in unrealistic scenarios and synthetic output[67].

Real testbeds are introduced as a solution for the simulation’s problems, where a miniature of a real life scenario is implemented. This gives the ability to test applications on native environments and evaluate the Quality of Experience (QoE). Running costs of real testbeds are very high in addition it suffers from multiple problems such as scalability, limitation on specific scenarios, very long preparation and running time, lack of large scale evaluation and difficulty of repetition, profiling and modification of an experiment.

To solve the problems of real testbeds, emulation concept is introduced where parts of the hardware components are emulated. The emulation data could be an output of a simulator, real traces or event-based generator. For example, NS-2 and NS-3[68] provide and emulation extensions that enables using simulated nodes as if they are real network nodes. However, this technique could lead to unrealistic results due to simulators’ overhead that is required to process the information[68].

Software emulators are usually provided by operating systems and hardware manufacturers to emulate several hardware profiles and supply visual tools on the developer’s desktop. This provides a simple yet effective way to validate an application, examine its look and feel and QoE. However, these tools are not suitable for VANET requirements, since they are not capable of emulating even simple wireless signal degradation between two moving nodes without changing the code of the applications under test. Moreover, they are not designed to support complex mobility or integration with urban mobility simulators. Unfortunately, there is no testbed or emulator that can handle the requirements for VANET's applications emulation and simulation necessities.

In this paper, we are taking the research of VANET testbeds one step forward by introducing a flexible VANET testbed architecture that fits the complex VANET requirements. Its flexibility lies mainly in these four concepts: 1) it is a layered architecture that is decoupled from specific sources of emulation data and make use of already existing real traces, network simulators, urban simulators and sensor data, 2) it can be deployed on real devices, providers' specific emulators and virtual machines, hence, enables the cooperation and ensure the maximum scalability, 3) it uses databases to operate which makes it very versatile to handle and modify simulation scenarios, 4) it incorporates industry state of the art operating systems, link emulation tools, and standard software engines to operate.

Our proposed testbed enables emulating sensor output (such as location sensors), complex road topology and mobility scenarios, signal strength, and wireless channel quality parameters (including jitter, bandwidth limitations, and packet loss). In addition, it natively emulates the wireless interfaces, which permits context-aware applications to use them as context providers and provides a realistic validation and evaluation environments.

In the following section we discuss the problem statement and requirements of a VANET testbed then we demonstrate the related work. In section 4.4, we present our proposed solution and discuss how it answers the VANET testbed's requirements, then in section 4.5 we explain testbed implementation and validation then we analyze the effect of the newly introduced caching entity on the testbed's performance. Finally, we present the conclusion Section.

4.2 Problem Statement and Testbed requirements

Due to unique characteristics of VANETs such as V2V communication, unpredicted driver's behavior, network fragmentation, road network topology and dynamic mobility, validating VANET applications is a challenging task. Moreover, the availability of embedded sensors, such as proximity, location and light, enables the development of complex context-aware applications. In addition, standard wireless interfaces can be used as a context provider as well as a communication interface, for example, in present time cell phones WiFi, Bluetooth, and cellular beacons are used to enhance the location service availability and accuracy.

Network simulators such as NS-2 even when combined with urban simulators such as SUMO, fail to evaluate the performance of such complex applications. Furthermore, using simulators there is no way to assess the user experience and application feedback. To cope with these requirements using a testbed is mandatory. However, there isn’t available feasible testbed that answers the demands of VANET applications.

In this section, we discuss the requirements of a VANET testbed. Firstly, we discuss the context awareness, then hardware and operating system requirements. Then we investigate the transparency, virtual device support, and hardware abstraction necessities. Lastly, peer to peer link emulation, repeatability and profiling, emulation data source independence and emulator overhead minimization are discussed.

4.2.1 Context Awareness

In the last five years, the difference between smartphones and desktops has changed from smartphones having limited functionality and being lightweight (in contrast to powerful and stationary desktops), to phones that are both context-aware and user aware (while desktops have remained powerful and stationary). At present, the main difference between a desktop and a smart device is the smart device's mobility and ability to interact with its immediate environment. This interaction is based on two main steps. Firstly, the device senses the environment, and subsequently, it changes the behavior of a particular application because of this environmental input.

Such real world context-rich smart applications require interaction between different context providers. The more context providers that a mobile application can access, the more functionality and application scenarios it can provide to the user. For example, the use of GPS as the only context provider for any location-aided application is not suitable for every situation since it is useful for only 4.5% the average uses in outdoor environments and is not be suitable for use in an indoor environment [69]. However, the addition of more context providers such as cell ID, Bluetooth beacons, and other sensors can improve localization services dramatically in an indoor environment.

Since the behavior of the context-aware application depends mainly on the surrounding environment, it is crucial to emulate the surrounding environment accurately in order to verify the context-aware application in an environment that is as realistic as possible.

For VANET applications, location information, neighbor wireless information and V2V wireless connection properties between vehicles are very crucial for an application to function in addition to other embedded sensor information in the smart device as shown in Figure 4.1.

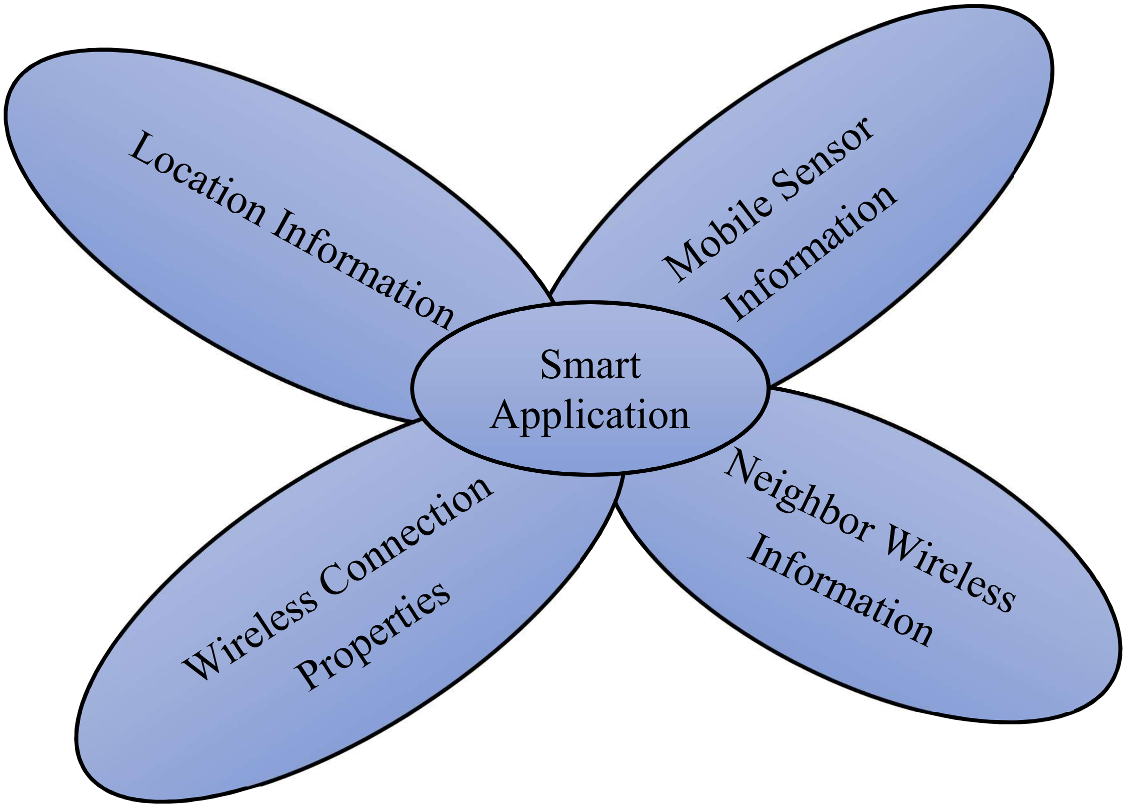


Figure 4.1: Smart application consumed information.

From a context-aware application point of view, context can be sensed by a user input, an external entity that shares the context or it can be automatically detected. An example of user input context awareness is an application that changes the device's profile with respect to the user’s schedule, for instance, it may modify the profile to a quieter "meeting" setting when there is a meeting planned for a particular time.

Context information sharing refers to the provision of context information by an external entity for the service and use of the smart platform. This external entity could be a partnered smart hardware platform or a service provider.

Examples of such partner context sharing can be an external location service which is available for partner users to obtain location data. Automatic detection of context, on the other hand, requires the available hardware interfaces to be on the smartphone for the device to take action.

Hardware interfaces that can be used as context-aware providers can be categorized as one of the following types:
1-Location providers (such as GPS sensors).
2-Wireless interfaces (such as Bluetooth, WiFi, etc.)
3- Embedded sensors (such as for proximity, light, accelerometer, gyroscope, temperature, etc.)
4-External providers (such as cloud-based and peer-to-peer context-sharing applications).
Wireless interfaces are used in the present-day smartphones to predict or enhance the current location. This is done by means of wireless beacons and the proximity and\or the triangulation with different wireless technology. For example, the Android location API is using WiFi access points' detected locations as location providers to enhance the detected GPS readings in the indoors environment or to decrease the number of times the application is polling information from the GPS onboard sensor to preserve the battery.

In D2D and WiFiDirect VANET application, location information of neighbors along with connection properties are key characteristics for an application to perform. Since scanning for neighbors in the proximity is not only dependent on the current location of the neighbors but also the wireless connectivity between them. Video on demand sharing [70] and pervasive gamming[71] using WiFiDirect and cloudlets[72] are promising sharing applications.

To the authors' knowledge, no emulator provide an extensive interaction among sensors for VANET and mobile applications.

4.2.2 Real Hardware Integration

Verification and validation of smart applications, especially VANET, requires using real hardware platforms. This gives the user the ability to exercise the application under test and evaluate the Quality of Experience (QoE) and the look and feel of the application.

In addition, the tester should be able to test different hardware to evaluate the performance of the application and trace any bugs.

4.2.3 Platform Operating System

In order to deploy and test applications context-aware applications, an operating system is required. Since the look and feel and QoE are vital aspects for an application emulation, these applications should run on its native operating system.

Ideally, the testbed should support all or at least most common operating systems that are used on smart platforms. However, this requirement is hard to accomplish since not all the operating systems are open source and some of them are not easy to extend.

If only one operating system is selected, this operating system should be one of the dominant smart platform operating systems, widely adopted by industry, open source and easy to extend.

There are known limitations of operating systems when deployed on virtual machines, such as the inability of natively using internal WiFi interfaces. This issue has to be taken into consideration.

4.2.4 Transparency

One of the greatest challenges of an emulation testbed is its transparency for the application under test. The application should be treated as a black box where neither modification to the source code nor special application plugins to be added.

For a testbed to be transparent, the user should not experience a difference when using the application on a testbed or real hardware.

The need for location data emulation is introduced in several early research based on Java 2 Mobile Edition (J2ME) applications and overriding the Location Provider in the J2ME Location API (JSR-179) by a data client for an external position data provider. In[73] the authors created custom location providers that read the RSSI from Bluetooth signals to assist the location, in addition, it can use a table of zone location boundaries for reasons of implementing indoor and outdoor location sensing.

However, the biggest problem of this technique is that the applications under test are location provider dependent that means a packaged application cannot be tested without modifying it to use the implemented provider or use a custom location implementation library.

Normally, the application under test is not available for us to meddle with and to change its code, so there is no way to modify the way an application consuming the sensors without emulating the sensor output.

4.2.5 Virtual Device Support

An ideal testbed should be able to support the communication among several environments. For example, in VANET we may need a hardware client in the experiment as well as a large number of virtual devices.

On the one hand, there is a need for a real hardware integration into the testbed as we described earlier to evaluate the QoE.

On the other hand, virtual devices are essential requirement to ensure the scalability and large network support. Imagine, having a VANET application under test that requires sending and receiving data to 100 devices; it would be impractical to use all clients as hardware.

Ideally, a VANET testbed should support several smart hardware platforms (e.g. smartphones, tablets), virtual devices that are running on virtual machines and support the communication with other provider-specific emulators.

4.2.6 Hardware Interface Abstraction

In order to ensure the scalability, the testbed emulated data should not be limited to only one interface. There are many types of research in the literature that generated an emulation testbed that is suitable for a specific wireless technology such as WiFi or Bluetooth.

Although a technology specific testbed could be simpler and more accurate, it is not easy to configure it to use newer technologies. For example, if there is an application that is designed to use Bluetooth for communication between devices, but newer wireless technology became available (e.g. Zigbee). Preferably, only the interface information and wireless properties are the ones need modification, and all the other information should remain the same including location, mobility and test case setup.

The same concept is used in network stacks, where the physical signal properties in the wireless interface are wrapped in the physical layer where network layer, for example, does not need to change its protocols.

4.2.7 Peer to Peer Link Emulation

Peer to Peer (P2P) link emulation is a crucial requirement for a VANET testbed. Since V2V communication may suffer jitter, delay, packet loss, and other network-specific criteria that affect both performance and QoE of the smart application under test.

In VANET, link emulation for V2V communication is challenging, since location, road topology, speed, and driver behavior affect not only the data flow between the two vehicles but the possible existence of a successful connection.

For example, for WiFi-Direct and LTE-A D2D communication, there is a scanning phase that has to run first to detect neighbor vehicles in the proximity. If the scanning phase were not successful, there would be no connection between the two vehicles. Results will vary depending on the application architecture and the time it requests the peer scanning.

P2P link emulation should be dynamic and easily configurable on each device. This gives the testbed user the agility to modify link qualities on each link for the application under test.

4.2.8 Repeatability and Profiling

One major advantage of using an emulator or a simulator is the ability to repeat the same experiment multiple times with slightly different parameters to measure the sensitivity of the application. There for a successful testbed, should facilitate repeating the same experiment multiple times and for multiple devices.

Profiling is another important feature of a testbed. The testbed should be able to store and retrieve specific situation profiles easily. These profiles include wireless network related, user experience related and mobility schemes.

In VANET, mobility pattern is a major requirement, where the testbed store several mobility profiles (e.g. downtown, highway and road topology scenarios). These scenarios do not have to be generated by the testbed, but at least it is stored and can be easily modified.

4.2.9 Emulation Data Source Independence

Emulation data source is a vital component in any emulator. Since this data will be used to emulate sensors, mobility, interface and network link properties.

The used testbed should not be coupled to a specific emulation data source. However, it should be agile enough to consume emulation data from different providers. This will ensure the adaptability of the testbed with future data sources.

The emulation data providers could be real traces, synthetic traces or results from other industry standard simulators. Ideally, the testbed user should be capable of without difficulty adjusting the emulated data to fit the test case requirements.

4.2.10 Emulator Overhead Minimization

Using an emulator resulting in overhead in performance and the environment when compared to real life experiments. A successful testbed should minimize the overhead usage so that it does not worsen the performance.

In the case of using emulated data, the time taken to retrieve the data and prepare to be consumed by the application under test is considered an overhead. Therefore, a testbed should be able to supply the emulated data to the platform in an efficient manner.

4.3 Related Work

In this section will demonstrate some well-known case studies of the implementation and research of successful network emulations. Note that this paper places emphasis on the software emulation approach since this approach is more realistic in its use as a cornerstone for a V2V testbed.

4.3.1 Sensor Data Emulation

The need for sensor data emulation, specifically location data emulation, is discussed in several studies based on J2ME applications and overrides of the “LocationProvider” in the standard J2ME JSR179 applications by a data client for an external data provider. In[73], the authors created custom location providers that read RSSI from Bluetooth signals to assist location sensing. It is also shown that these custom location providers can use a table of zone location boundaries to implement both indoor and outdoor location sensing. A map GUI is suggested and implemented[74] to allow for users to input locations through keyboard and mouse actions.

Chen, Y., et al. outlay a full architecture of mobile middleware for Location Based Services (LBS) in [75], which is called a location operating reference model (LORE). It handles important aspects of location emulation such as query processing of locations, tracking, location servers, database design and Application Program Interface (API) to be used by mobile applications. The architecture can be utilized in an enterprise manner, to provide location and to track in both pull and push modes.

Open source Location API (OpenLAPI)[76] provides an open source implementation model for JSR179 that enables the abstraction of location providers and location emulation. Using OpenLAPI enables J2ME developers to provide emulated location input, for example from network socket-based location provider clients.

All these location emulation attempts face similar problems since they are based on JSR179 standard, which is not suitable for simulating dynamic scenarios as it was built primarily to handle resource-critical devices[77]. Furthermore, JSR179 does not provide support for new location context requirements such as geo-coding, landmarks and map user interface[77]. In addition, they are location provider dependent: that is, a packaged application cannot run on a simulator without being modified to use an implemented provider.

Campillo-Sanchez, P., et al. in Simulation Based Software Development for Smart Phones [78], introduce an agent-based 3D editor that can be modified using a keyboard and mouse in a game-like technique to test complex context-aware application scenarios. The tool uses an agent-based simulator to emulate different context information such as simulated accelerometers, temperature sensors, a map of an indoor floor plan with doors, and so forth. The tool allows the tester to navigate with the phone in a way very similar to that of an interactive game. This research focuses on both location emulation and sensor interaction emulation; however, it does not address wireless signal emulation and usage.

4.3.2 Data Link Emulation

The Open Access Research Testbed for Next-Generation Wireless Networks (ORBIT)[54] is a hardware-based network emulation testbed. It consists of 802.11 wireless nodes that are arranged in a two-dimensional grid. Each node contains wireless interfaces and Ethernet ports. For emulation purposes, a framework is developed to enable users to define a scenario, then deploy the scenario on nodes, run experiments and finally obtain the results. ORBIT is very useful for conducting analyses of new protocols; however, it lacks the ability to emulate complex mobility and context-aware applications. In addition, the test environment is not controllable, since it is based on real hardware interfaces.

PlanetLab[55] consists of a set of virtual machines that are scattered and deployed globally. These virtual machines are divided into smaller slices geographically distributed where they can be accessed to test applications globally. PlanetLab provides a useful resource for testing internet-enabled applications, however, it is originally designed to handle wired networks and it is not suitable to handle wireless ones.

DummyNet [79] is a simple, lightweight and efficient traffic shaping tool. It works as an intercepting filter to add delay or jitter, and to limit bandwidth. The implementation consists of less than 300 lines of code and is available under a FreeBSD license. DummyNet is one of the first attempts to bridge the gap between simulation and emulation by providing traffic-shaping functions seamlessly to an application under test.

NetShaper[53] is similar to Dummynet in that it emulates link layer properties, but it was also developed as a kernel loadable module. The main difference between NetShaper and DummyNet is NetShaper's support to dynamic network configuration. NetShaper contains a configuration utility application that runs on the user space in order to control the link layer emulation's parameters. It communicates with control nodes to receive control commands using UDP messages. The control node is configured to change the network configuration in a dynamic manner.

Simple Emulation of Delay and Loss for Ad Hoc Networks Environment (SEDLANE) [23] is an extension for DummyNet in which NS-2 simulations are used as a source for delay and packet loss information that DummyNet requires in order to operate. The idea such use of NS-2 data is to simulate complex scenarios that would be very difficult, if not impossible, to replicate in real environments. We are sharing the same methodology as SEDLANE where network simulator outputs are used as an input for a link emulator, allows for the simulation of complex traffic scenarios, obstacles and signal properties in different environments.

KauNet [24] can be considered an extension for DummyNet. It provides the ability to create traffic shaping profiles that can be data-driven and discrete data-driven for DummyNet. These configuration files ensure the reproducibility of the traffic shaping experience. Moreover, KauNet provides the ability to introduce bit errors, which is not supported by DummyNet. There is a slight performance difference between KauNet and DummyNet, an experiment performance analysis of which is studied by Lubke, R., et al. in [25].

NISTNet[26] provides a robust traffic shaping tool. It is available in Linux kernel 2.4 and is patched to higher kernel versions. Similar to DummyNet and KauNet, NISTNet provides tools to throttle bandwidth, add packet delay and apply jitter. Moreover, it provides a GUI to configure traffic shaping parameters.

NetEM[27] and traffic control (TC) tools are widely used and integrated into Linux kernels. They are very similar in concept to NISTNet. However, NetEM and TC are superior in terms of the number of operating systems that support them. For example, Android framework currently supporting the TC and NetEM commands, along with most Linux distributions (with patching required).For an experiment-based performance analysis comparing DummyNet, NetEM and NISTNet, refer to [28].

Conchon, E., et al. W-NINE: a two-stage emulation platform for mobile and wireless systems introduce W-NINE[29] as a multistage network emulation framework. Simulation for a preconfigured network with the nodes and their mobility can be conducted using Simulator for Wireless Networks Emulation (SWINE) in order to predict jitter, bandwidth, and other network characteristics. The output from the first stage is then used as input for a KauNet [24] controller router that applies these effects upon the nodes participating in an emulation. The idea of using two-stage simulation is time consuming but very effective in reducing the time that a simulator needs to process the information. However, SWINE has not been updated and its suitability for supporting and integrating with other road traffic mobility’s simulators is not clear.

Considering the above emulation approaches, the best recommendation for a link emulation tool is NetEM\TC; since it is supported by most Linux distributions and more importantly Android, as compared to other available emulators. NetEM\TC is flexible: it can also operate as a standalone on each device and since it is Linux\Android based, it is easily used with Android hardware, Android emulators, Android virtual machines, or Linux based desktop clients. Moreover, it supports all required elements for link emulation such as bandwidth limitation, jitter, packet dropping, packet duplication and packet corruption [28]. The idea of using multistage emulation (as does W-NINE[29]) is praiseworthy, but overall, a more generic form of a test platform is more functional, because it can integrate with different traffic simulators and utilize captured real life wireless traces.

It would also be worthwhile to use network simulator output's traces, such as the output NS-2 in the case of SEDLANE, as input data to link emulators. However, in order to obtain results, we recommend integrating road traffic and driver behavior using road traffic mobility's emulator.

4.4 **Proposed Solution**

In this section we will discuss the proposed solution and how does it satisfy the requirements. First, we will discuss the general idea of the testbed, then we present the testbed architecture and its components. Finally, we will discuss the link emulation method selected and how does it work.

4.4.1 General Idea of the proposed Testbed

With the aim of satisfying the requirements of real hardware Integration, transparency and virtual device support, we selected the idea of modifying an open source operating system for our testbed.

By modifying and extending an operating system, we have the ability to wrap all the connections and sensors providers to get the content from external sources as will be discussed in the next section.

Moreover, modifying and existing operating systems gives us the ability to use the already built tools and supported virtual and real hardware platforms. As shown in Figure 4.2, we can deploy the modified operating system on actual hardware, an emulator or virtual machines.

Hence we are wrapping the network provider; we can emulate any network connection types and changing the properties of these connections. For example, an operating system could be deployed on a virtual machine. However, the virtual machine host does not have a WiFi Interface, this normally causes the operating system and hence the application under test to fail to use its WiFi capabilities. However, since we are wrapping the WiFi connection itself, we can emulate a WiFi interface with all the QoS, neighbor wireless devices, and link QoS.

The synchronization of virtual machines' time is a hot topic of research, even to avoid a fraction of a second of difference. However, in our approach, we do not face the virtual machine synchronization; since we control the sequence of events by storing a relative timestamp for each sensor data (e.g. GPS location) in a central location. For instance, if virtual machine A's time is not synchronized with virtual machine B's time, this would not affect our testbed, since the first location for virtual machine A will be available after the configured milliseconds of the start of the application, and the same for virtual machine B.

In the configuration application, each virtual machine can select the starting offset of milliseconds (e.g. Device A can choose to start from 2 milliseconds). To ease the offset of start; simulation sensor data can be configured to be constant for the first 1000 milliseconds to allow for the caching mechanism to poll future sensor data.

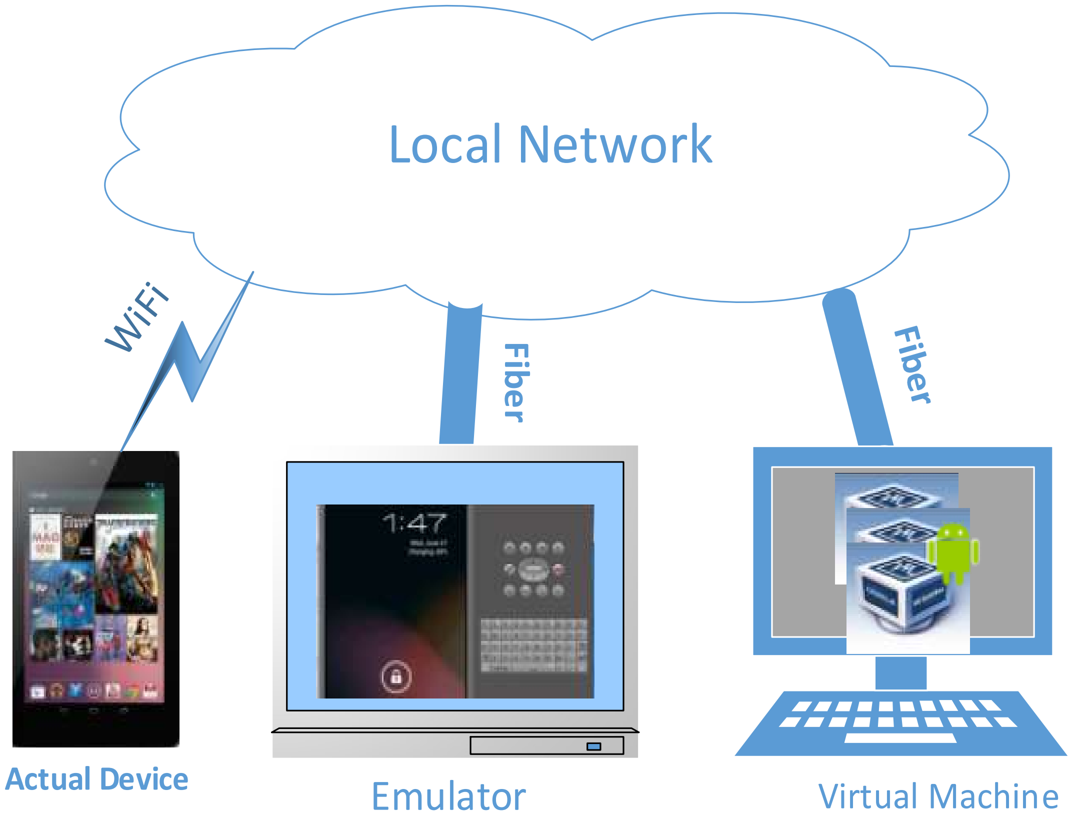


Figure 4.2: Overall Design

4.4.2 Testbed Architecture

Supporting multiple sources of emulation data is one of our top priorities for the proposed testbed. In order, to satisfy the requirements of profiling, repeatability and emulation data source independence, we adopt using a database concept to store the emulated data.

We adopt multistage emulation, where mobility and wireless traces are generated first before getting stored in the testbed's database to be consumed by the application framework when the application is ready to be tested as shown in Figure 4.3. The emulated data that are available from several sources such as real network signaling traces, road network mobility, emulated sensor data and wireless network simulator output are converted using the data provider mapping layer to match our database data structure and then stored in the database.

We extract the key performance wireless indicators such as jitter, packet loss, delay and available bandwidth from the wireless network traces instead of using the raw wireless network traces. Using key performance indicators reduces the amount of data stored in the system's database as well as gives the tester the agility to modify these values. Moreover, this approach is representing the QoS that is sensed by a higher layer application.

Although this extraction method is not giving very accurate emulation of the particular case wireless situation, it fits well wireless emulation for evaluating QoE for higher layer applications under test and makes the profiling of wireless properties based on specific QoS metric (e.g. delay variation) much easier to realize.

It worth to mention that, our scripts can be modified based on the wireless simulator used, and it is not coupled with a specific simulator. For the reasons of validating our testbed, we developed modules for NS2 trace parsing to extract wireless key performance indicators. We have also developed parsing scripts for SUMO traces to extract GPS coordinates.

There are two reasons behind selecting NS2 and SUMO, the first both of them are open source, and the second both of them are widely tested and accepted by the research community for simulating VANET applications.

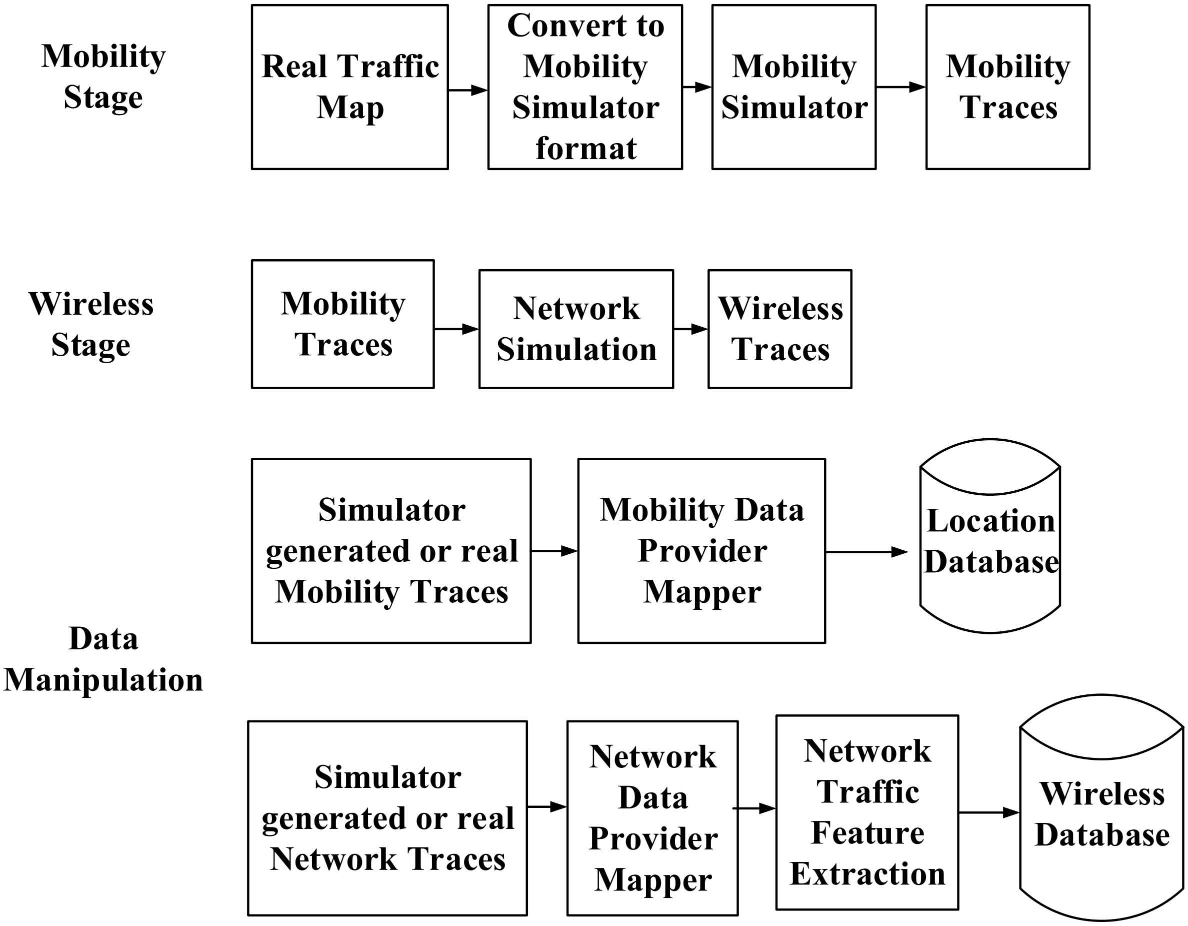


Figure 4.3: Multistage Simulation and emulation

As shown in Figure 4.4, our testbed is based on three main subsystems: 1) the inputs, 2) the core and 3) the client framework, which are discussed in details in this section.

In the core subsystem, we use databases to store emulated data. Using a database facilitates insertion, manipulation and retrieval of emulated data in a standard and efficient manner. In addition, SQL query language is very easy to understand and is widely adopted by the industry.

We use Java, Spring Framework, and shell scripting technologies to build multiple data provider mappers such as SUMO and NS2. Using Java ensures the portability of the program, yet using Spring Framework provides the ability of declarative programming and injection of data provider mappers automatically.

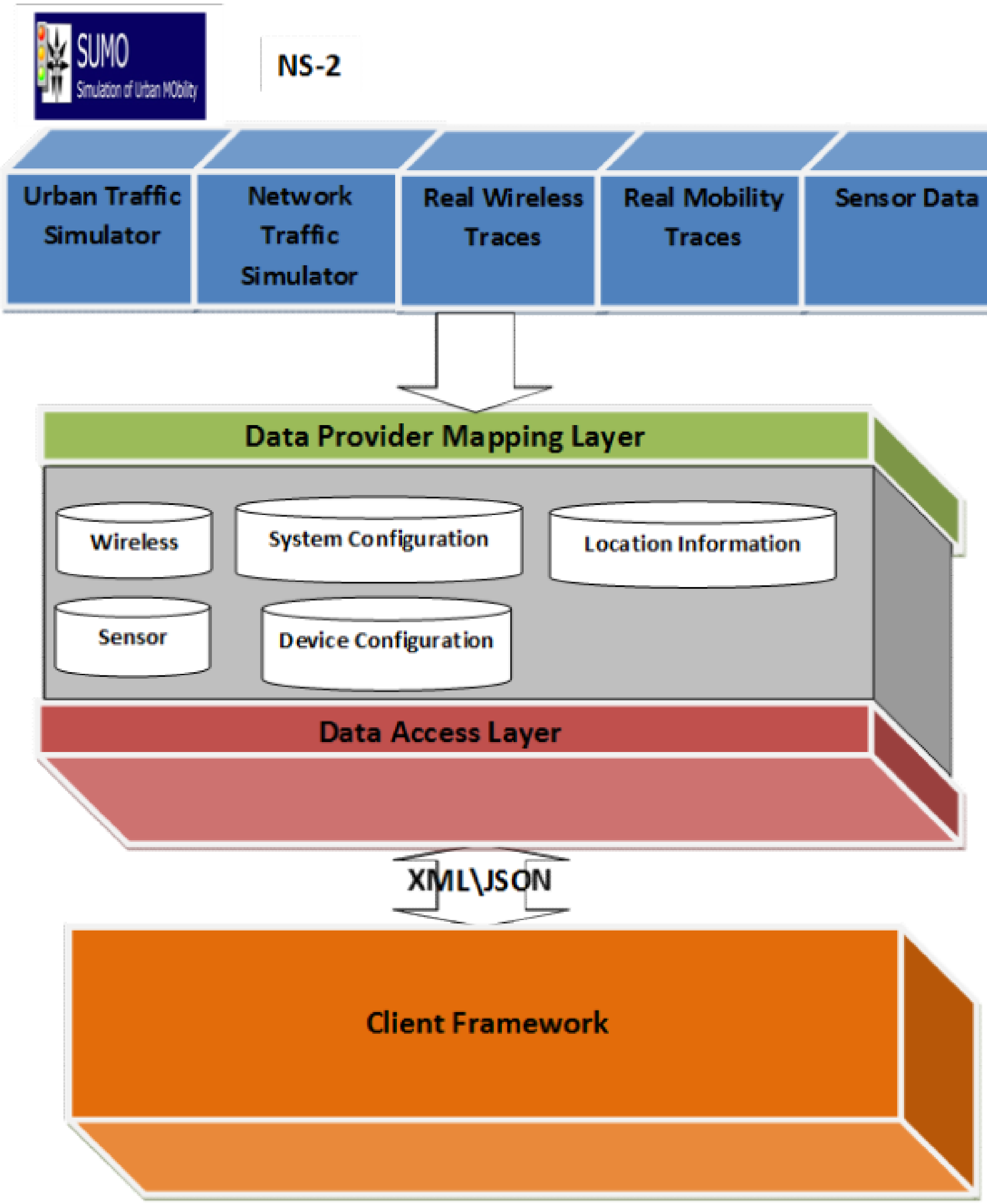


Figure 4.4: Overall Testbed Architecture

We categorize the main inputs for emulated input into four categories mobility and location, wireless, and system configurations. In this section, we discuss the rationale and the functions and the conversion of each category.

A Mobility and location data

For VANET mobility and location of each vehicle is a crucial feature for the application under test. The mobility emulated data could be an output of a mobility simulator (e.g. SUMO) or from reallocation traces (e.g. GPS traces).

From the application under test point of view, the location of the current device, as well as the locations of all the neighboring wireless devices, is necessary for the peer to peer communication. Hence our design is focusing on providing the location of each of vehicles for the application.

The location of each vehicle is stored in the format of longitude, latitude, altitude, provider type and timestamp. We selected this format mainly because it is commonly used in GPS and mobility simulators and there are geospatial databases that can natively optimize the location data.

The Data provider mapping layer is designed in a modular way, where each simulator, emulator or traces provider can build its own mapper to extract the data and convert it to the database scheme. It worth to mention here that the location information of neighbor vehicles are not known to each other at the time of testing the application. However, this data is used by the emulated interfaces to facilitate the peer to peer connection.

For example, if a vehicle wants to use WiFi Direct technology to scan for other vehicles in the proximity, the location information of the neighbor vehicles along with the wireless properties are used to determine if such a connection is possible or not and the expected candidates in the scanning phase.

Figure 4.5 shows a query step that is performed by our WiFi wrapper to search for nearby access points (APs). The query is based on the scanning device id and the simulation point of time where these APs are in range with the scanning device.

The result list of APs will be returned to the scanning device with its availability period, to guarantee the unavailability of these access points when the scanning device is out of range. This query involves joins with device tables to indicate the maximum range and location table of the current device to indicate its location with respect to the APs'.

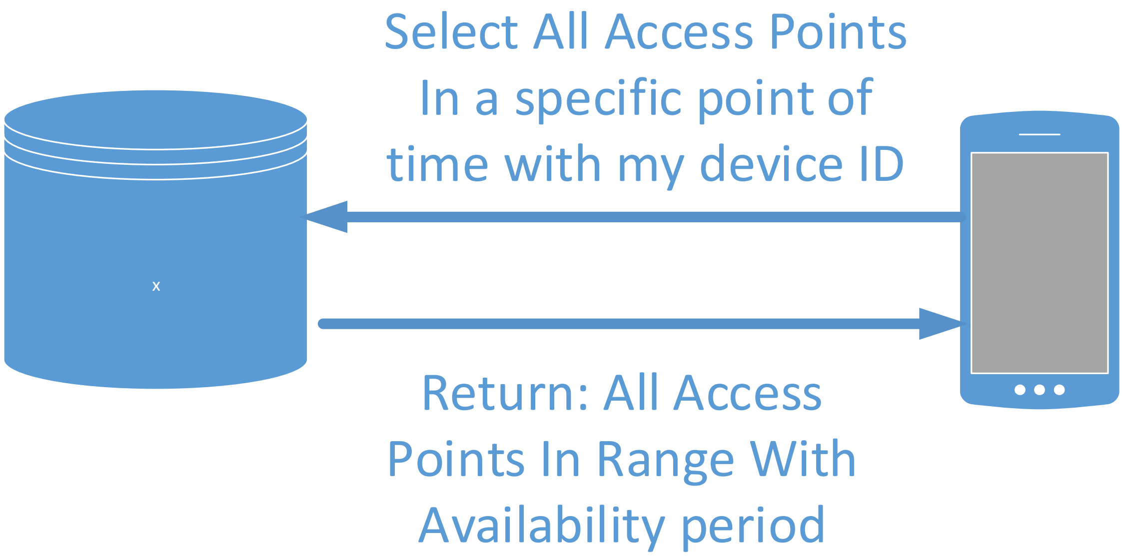


Figure 4.5: Part of Wireless Scanning Query

There are multiple many to many relations between the location database and the wireless database due to the fact that the location of access points and vehicles are essential for the wireless connection. It worth to mention that there is an option to add an obstacle configuration where the two vehicles are in proximity, however, the wireless connection between them is not available.

B Wireless

VANET smart applications depend on wireless technologies to communicate with each other, therefore emulating the wireless properties are essential for the testbed to function accurately.

The main reason behind building this testbed is the experimentation with smart and context-aware applications under test. Therefore, the wireless information that we need to store and consume by the testbed are these that can be sensed with higher level applications.

For example, a higher layer application that broadcast information is neither concerned with the physical layer propagation nor the data link layer protocol used by the wireless interface, however it is concerned with bandwidth, jitter, delay, and packet loss rate which can degrade application performance.

Instead of storing the wireless readings provided by simulators, emulators or real traces, we extract the Key Performance Indicators (KPI). There are many reasons behind this selection are mainly 1) the data provided by real traces although it is accurate it is not configurable to a specific scenario, 2) we need the ability to profile specific wireless traffic scenario, 3) simplifies the process of manipulating the wireless experiment.

We identified Signal to Noise Ratio (SNR), delay, packet loss and data Bandwidth (BW) to be the minimum required KPI for our testbed to function accurately. Delay and BW are more observable than others by the application under test, since the context-aware applications that use the wireless capabilities to communicate to each other are affected by how long it takes to send and receive data blocks.

To abstract the BW estimation, we store the BW as a percentage of the maximum data channel bandwidth. This can be beneficial to test applications different BW profiles. For example, if an application is requiring freezing all the process until specific data is sent or received through a wireless interface, it may not perform well if only 10% of the BW is available.

C System and Device Configuration

In the device configuration database, we store the default information for each device. This information can be the available wireless interfaces, location providers and sensors on this device. This information can be used in different experiments, since it is creating profiles for the devices used in the experiment no matter how will these devices will be utilized after in the emulation.

Each virtual, real or emulated device that is employed by the experiment is mapped in the testbed. The device information is then associated with its corresponding identification in the wireless and location database. For example, in NS2 trace a node named "Node1" after insertion in the database it will get a generated Public Key (PK) this PK is used to link the device configuration "Device 1" to "Node 1".

Each device contains multiple wireless interfaces where each interface has a name, a type and a coverage radius in addition to the detailed information of each wireless interface table, such as mac_address, address, port and all the other features required by a wireless interface.

Other system configurations such as default server port, default starting simulation time, login information, display formats are also stored in these tables.

4.4.3 Client Framework

The proposed client framework is built on four layers namely: Adapter Layer, Manager Layer, Caching Layer, and Configuration layer, as shown in Figure 4.6. In this section, we will illustrate the function of each layer.

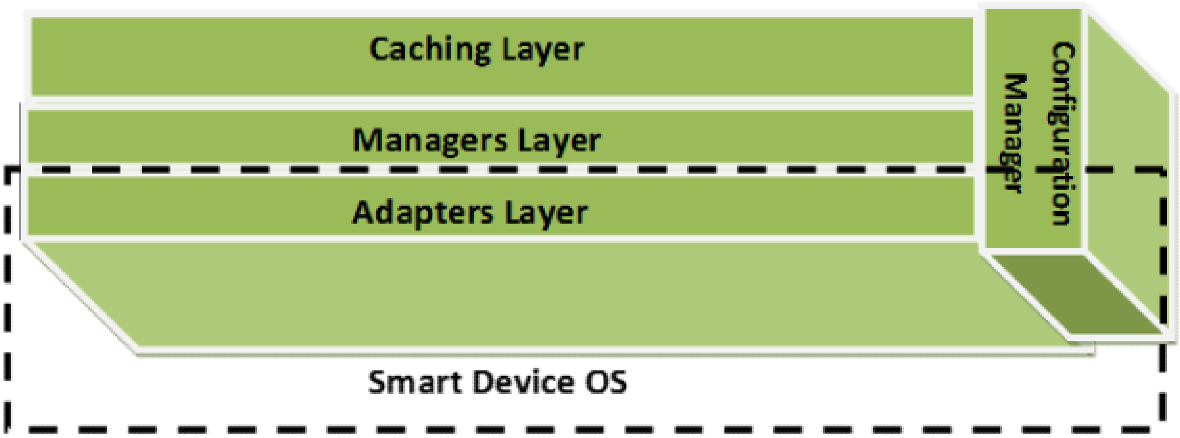


Figure 4.6: Client Framework

A Adapters layer

This layer communicates directly with a smart device's native API. As its name suggests, it adheres to the adapter design pattern. Third-party applications can use the emulated interface exactly the same way as the native OS interface while the implementation is different. The adapter layer contains a wireless adapter, a sensor adapter, location adapter, and a P2P adapter. The wireless adapter provides functions that are exposed in the native OS such as the retrieval of signal strength and neighbor access points. The location adapter provides information about location, simulation source and accuracy.

B Managers Layer

The manager layer is responsible for managing different adapters. For example, WiFiManager is in charge of connecting to the WirelessAdapter in the adapter layer in order to retrieve wireless information and properties. It is also responsible for applying the wireless properties for communications, for example by applying the wireless delay to the data received from a specific node, and based on simulated context information.

C Caching Layer

The caching layer is responsible for caching context information on a local smart device. One major problem with retrieving context information from the database is that it will add extra delay to the applications' functions. For wireless communications, this delay could be over 250 milliseconds for each context information pull, which is not useful for the simulation of real-time wireless applications. The solution here is for caching to be used to retrieve a “chunk” of data and store it for future use. Hence, a chunk of context information can be pre-cached before the start of an actual simulation. Our experiment showed that this caching technique reduced the time for context information retrieval to below one millisecond.

D Configuration Layer

Configuration Layer is responsible for configuring the parameters for all the proposed layer. The configuration layer is retrieving the initial configuration from the context provider database as well as user specific input on the configuration screen on the device. For example, the polling rate and the amount of data to be cached are two different configuration parameters that are set by in the caching layer.

In Figure 4.7, we show a sequence diagram of an Android smartphone application that requests the list of peers in the WiFi neighborhood using WiFi Direct capabilities in the WiFi interface.

In order for an application to get the default WiFiP2PManager, it has first to use the “getSystemService” method. In our custom Android, we override this method to change its behavior and return the emulated WiFiP2PManager instead of the default Android one. The emulated WiFiP2PManager enables us of emulating all the functions provided by the WiFiP2PManager and using emulated data as discussed in the previous section.

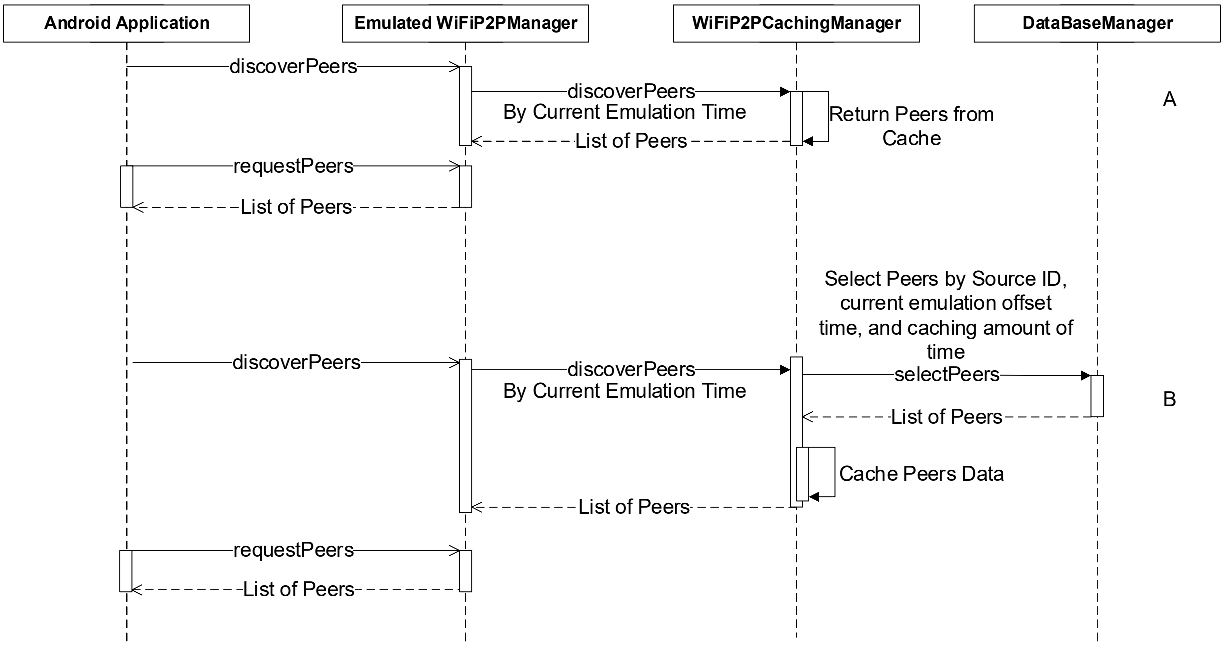


Figure 4.7 : Testbed Client Framework Sequence Diagram.

After getting the WiFiP2PManager, the application call “discoverPeers” method to scan for available peers in the wireless domain. “discoverPeers” is an asynchronous method; so that the application does not wait for a response. However, when the “discoverPeers” operation is completed, it calls a callback function to set the status of the operation as a success or fail.

Once the “discoverPeers” method is successful, the user will be able to call “requestPeers” method that is already populated with the list of peers in the proximity.

In Figure 4.7 situation A, there is a cache hit where the requested information is still in the cache manager and can be retrieved locally. However, in situation B the list of peers requested is not available in the cache, so the data retrieved is from the database manager which connects to the data access layer in the server to run a query and return the results. Once the results are received, they are cached and then only the requested information is returned when the “requestPeers” method is called.

4.4.4 Peer to Peer Link Emulation

Since there is no support for wireless channel emulation with mobility, we must apply jitter, delay and packet loss for the data communicated between each peer. We envisioned throttling and modifying the emulated channel only as the sender side as shown in Figure 4.8.

The channel quality data is retrieved by the adapters and managers from the database and applied to all the data at the source. This way we can distribute the load of applying QoS on all the devices on the testbed and not on a single point such as a dedicated central QoS emulation server or similar single point of failure designs.

Our recommendation for a link emulation tool is NetEM\TC; since is supported by most Linux distribution and more importantly Android if compared to other available emulators. Moreover, it supports all the required link emulation requirements such as Bandwidth limitation, Jitter, Packet dropping, Packet duplication and Packet corruption [28].

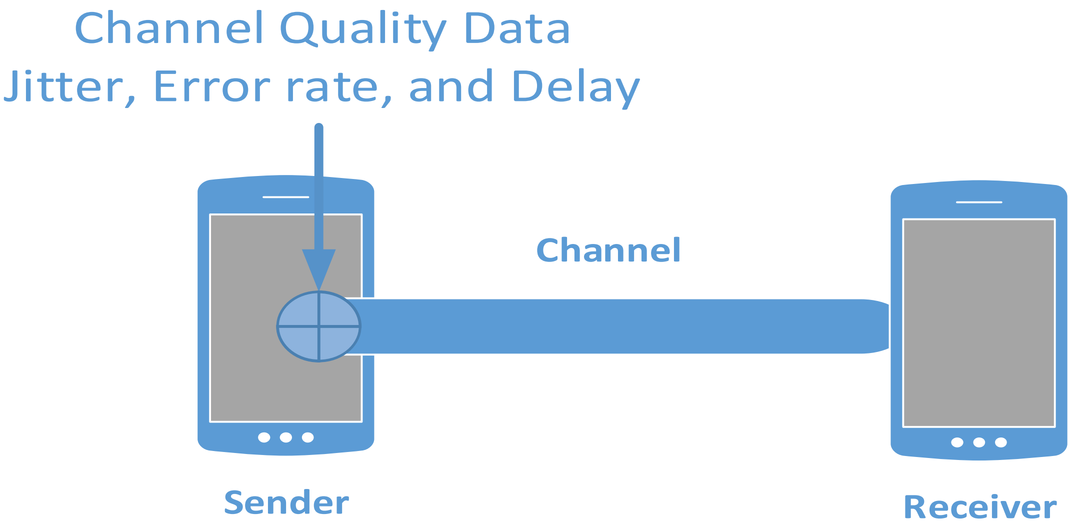


Figure 4.8: Link Layer Emulation

Moreover, NetEM\TC can work as standalone on each device, and since it is Linux\Android based, it can be easily used on Android Hardware, Android Emulators, Android Virtual Machines, or Desktop Linux based clients.

4.5 Performance Analysis and Implementation

In this Section, we discuss the operating system we selected for the implementation. After that, we present our testbed configuration application and applications that are developed to validate the functionalities of WiFi connection, WiFi Direct P2P and location emulation on our testbed. Finally, we conduct a performance analysis of using the newly introduced caching layer.

4.5.1 Operating System for Testbed

We selected Android to be our testbed operating system. With more than 1 billion active users, a significant number of supported devices. Supported devices include hardware, virtual machines in addition to the embed emulators.

Android platforms provide a well-structured and layered operating system that is open-source and natively built with the Linux kernel. Open source software gives developers the opportunity to efficiently enhance, customize and rebuild the source code of the operating system. In addition, the Android's component-based architecture - which uses managers to handle the sensors and connection interfaces - makes the creation of new providers easier, which can provide the simulated data needed for a particular study.

The Software Development Kit (SDK) along with Android Development Kit (ADT) plug-in (which is compatible with Eclipse Integrated Development Environment) provides an easy to use API for the development of context-rich applications. These kits also serve as a powerful development editor to develop software more efficiently. These kits also come with an Android Emulator that can be configured to emulate different Android hardware platform profiles (tablets, cell phones, memory, CPU, etc.).

One of the major advantages of Android over other closed source mobile software platforms is the availability of a virtual machine for building different Android hardware profiles. Project Android X-86 [80] is an open source project that aims to build custom Android images that can run on X-86 processor architecture (that is, normal PC) in place of specific Android hardware.

This project enabled two important uses of Android. It allowed PC users to run Android natively on their PCs as a primary OS or as a dual boot system with Android and a different OS. 2) It also enabled the creation of virtual machine images that contain specific configurations of an Android image and can run a virtual machine host on top of an OS. For example, Project Android x-86 provides Android images for Android version 4.0 version, which can run on virtual machine providers such as VirtualBox[81].

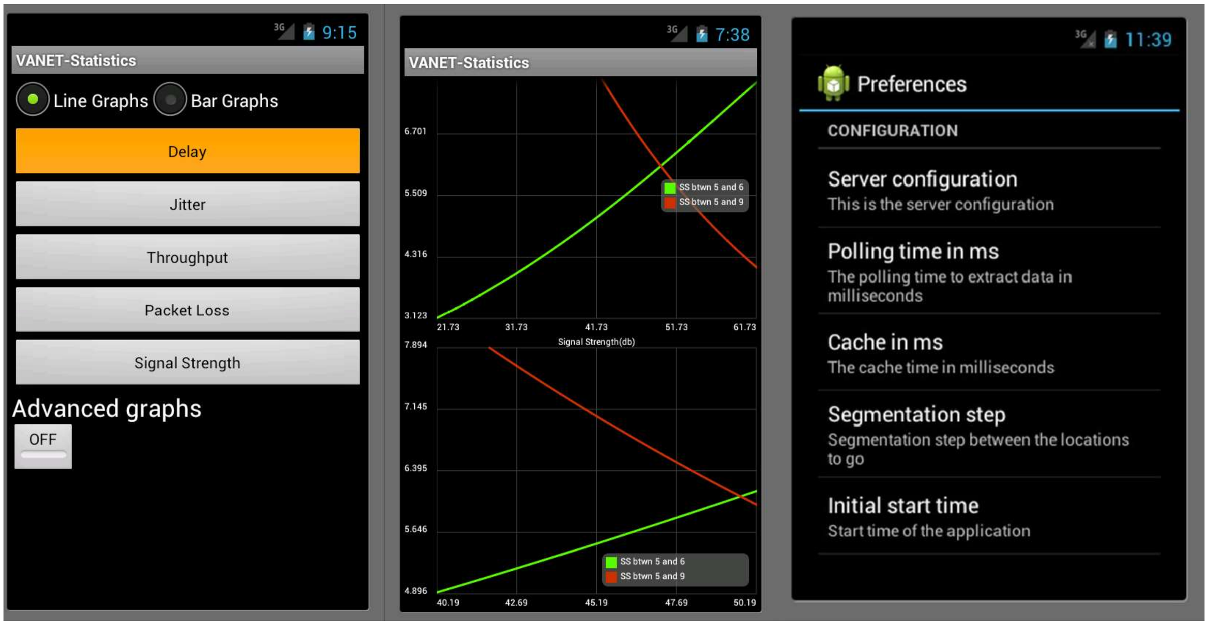
One challenge that faces Android emulators is the rapid and continuous development of various wireless technologies. For example, ten years ago WiFi was not available on most smartphones. Bluetooth soon started to be a common feature, and at present, WiFi is on almost all smartphones and is used by many smart applications. In the near future, it is expected that new wireless interfaces such as LTE-A D2D and conjugative radio technologies will be added to smartphones. However, most simulators and emulators on the market do not provide wireless support due to the sheer complexity of its implementation.

4.5.2 Configuration Application

This configuration application is vital for the testbed's operation where all the emulated data flow to the testbed's hardware is controlled. As shown in Figure 9, we assign the IP address of the server in the server configuration field. The configurable IP enables the decouple of the testbed device from the emulated data server. In addition, this can be very useful in load balancing and server profiling.

In order to control the caching, "Polling time in ms" and "Cache in ms" field are provided. The first parameter is used to indicate the frequency of polling the emulated data from the server (e.g. every 400, 600, 1000 ms). However, the second parameter is used to identify the maximum caching to be used on the device. These two parameters are very useful in limiting the storage of the information on each testbed node.

A simple smoothing algorithm for location data is implemented where the data between emulation steps are interpolated. For example, suppose there are only two emulated location information data and the time difference between them is 2 seconds. If the tester would like more data in between, by configuring the minimum time gap in the "Segmentation Step" field, the testbed will be able to emulate the environment more realistically in the event of lacking emulated data.



Server configuration
This is the server configuration

Polling time in ms
The polling time to extract data in milliseconds

Cache in ms
The cache time in milliseconds

Segmentation step
Segmentation step between the locations to go

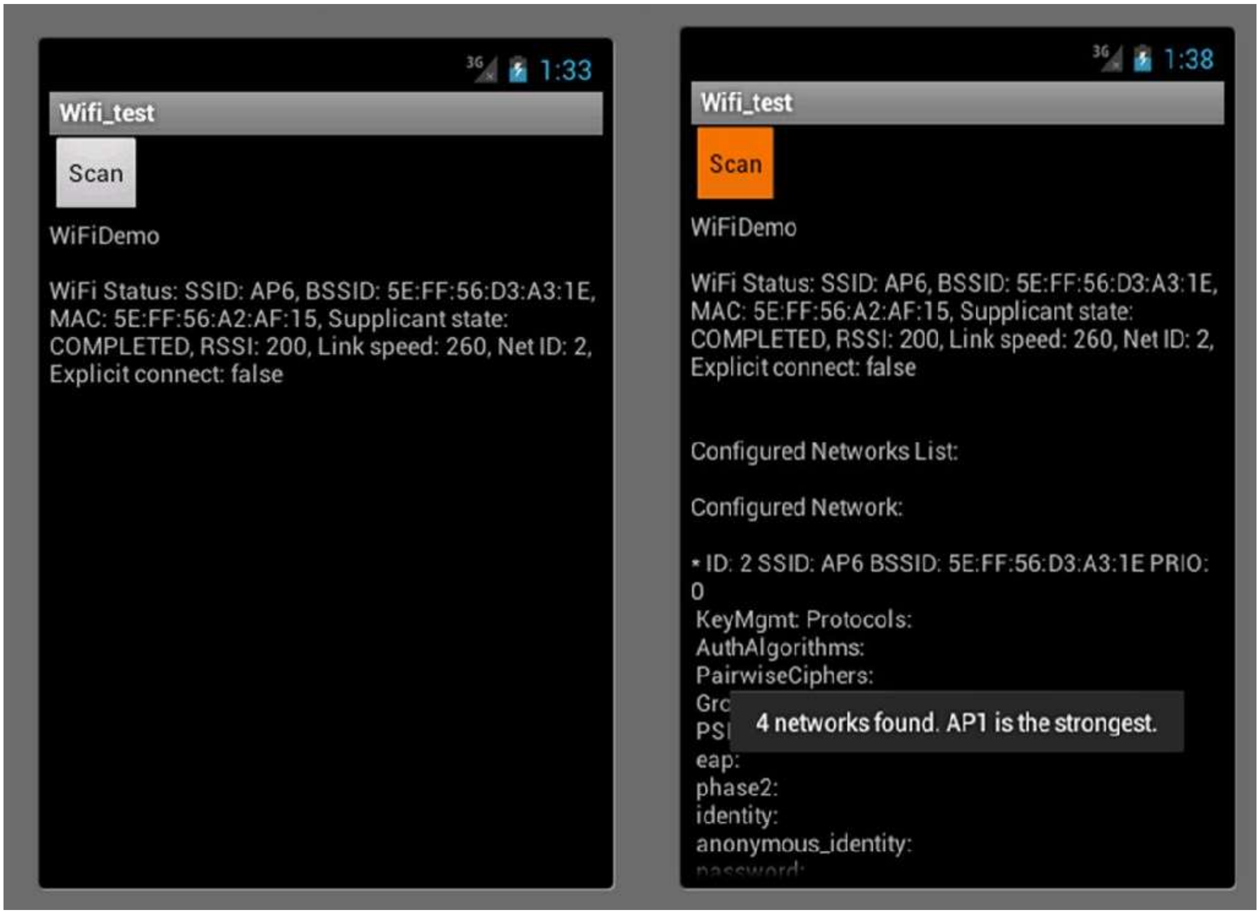
Initial start time
Start time of the application

Figure 4.9: Testbed Configuration application

In order to present the data polled by the testbed, we introduce a demo application that lets the user select a quality metric (i.e. delay, jitter, throughput, packet loss, or signal strength) and draw a simple graph for this data as shown in Figure 4.9. The demo application gives the user a quick tool to check that the testbed was able to poll emulated data from the server successfully. It worth to mention that due to the large number of attributes that can be emulated on a device, we only select a sample from this data to be displayed.

4.5.3 Emulated WiFi Evaluation

One of the main benefits of this testbed is enabling the WiFi interface emulation. We developed a simple WiFi demo application that is based on a tutorial. The application simply conducts a scan for surrounding WiFi access points, retrieve its properties and select the strongest one based on the signal strength as shown in Figure 4.10. This application validates that the emulated WiFi manager is working effectively in polling all the emulated data. It worth to mention that we did not use fabricated functionality to develop this application, just the standard API.



```
WiFiDemo

WiFi Status: SSID: AP6, BSSID: 5E:FF:56:D3:A3:1E,
MAC: 5E:FF:56:A2:AF:15, Supplicant state:
COMPLETED, RSSI: 200, Link speed: 260, Net ID: 2,
Explicit connect: false
```

```
WiFiDemo

WiFi Status: SSID: AP6, BSSID: 5E:FF:56:D3:A3:1E,
MAC: 5E:FF:56:A2:AF:15, Supplicant state:
COMPLETED, RSSI: 200, Link speed: 260, Net ID: 2,
Explicit connect: false


Configured Networks List:

Configured Network:

* ID: 2 SSID: AP6 BSSID: 5E:FF:56:D3:A3:1E PRIO:
0
 KeyMgmt: Protocols:
 AuthAlgorithms:
 PairwiseCiphers:
 Grc
 PSI
 eap:
 phase2:
 identity:
 anonymous_identity:
 password:
```

```
4 networks found. AP1 is the strongest.
```

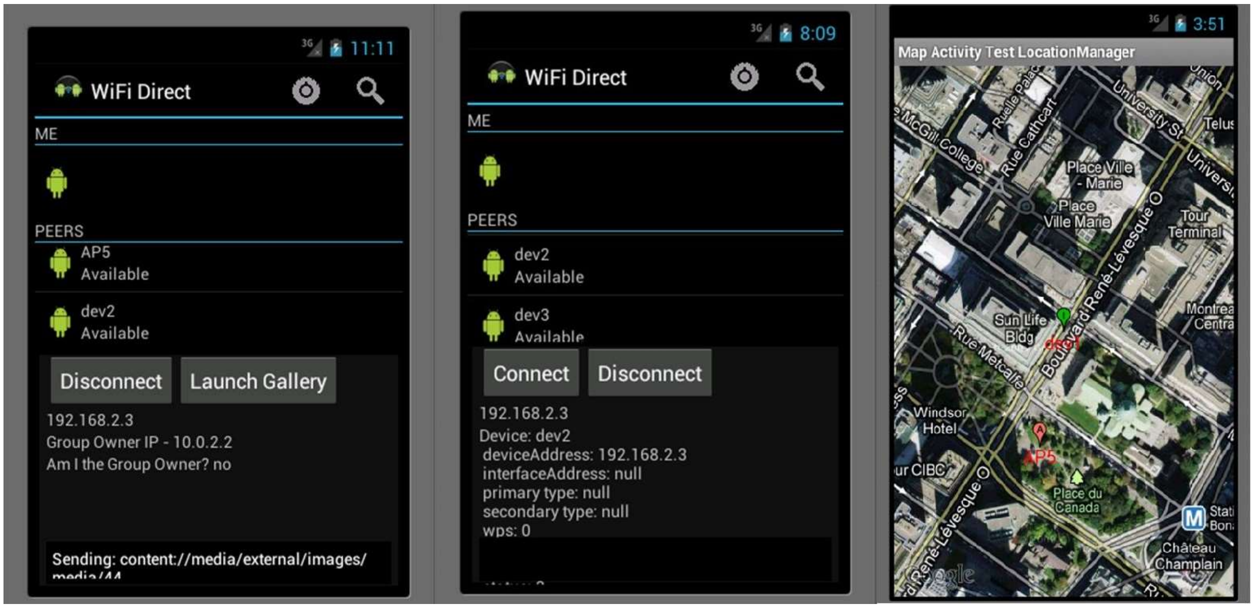
Figure 4.10: Emulated WiFi Application Demo

4.5.4 Emulated WiFi Direct

P2P communication is a vital feature that our testbed is supporting. In order to demonstrate the ability of our testbed to emulate P2P communication, we developed a WiFi direct demo based on sample applications provided by the ADT. This demo can scan for peers in the emulated communication proximity as discussed in Figure 4.7. Once the scanning list is ready, the user can select to connect to a peer and act as either a server or client to transfer a simple image. The emulated channels parameters govern the image transfer rate and quality are discussed in the previous section.

Since the mobility of the current node as well as the peers' mobility of peers, we developed a map application to display the mobility of the device under test on real maps as shown in Figure 4.11.

This application proves that location and WiFi Direct emulations are working successfully and the client framework is capable of communicating with the core subsystem and inject the emulated data.



192.168.2.3
Group Owner IP - 10.0.2.2
Am I the Group Owner? no
Sending: content://media/external/images/media/44

192.168.2.3
Device: dev2
deviceAddress: 192.168.2.3
interfaceAddress: null
primary type: null
secondary type: null
wps: 0

Figure 4.11: Emulated WiFi Direct application with visual mobility

4.5.5 Caching Layer Performance Analysis

The application prototypes discussed in the previous section show the ability of out testbed to emulate WiFI, WiFi Direct, and location-based applications. However, one major overhead of a testbed is the time taken to fetch and process the emulated data. In this section, we discuss the effect of the caching entity on the testbed performance.

We use the WiFi direct application discussed in the previous section to evaluate the effect of the caching entity on the emulated location manager. We used the configuration application to change the polling interval and the caching amount used. The emulated experiment is designed as a moving car in a downtown and continuously scanning for peers. We added logging functions to the location manager.

We calculated the average time fetching location information from the server each time it is requested. In order to avoid biased data, the experiment runs for 10 minutes.

We selected 400 ms of polling rate as an average polling rate. However, other polling rates are studied as well with similar results. We run the deployed the testbed O.S. on the Andriod emulator provided in the ADT.

As shown in Figure 4.12, the average time that is taken by an application to receive the location on a native device is about 55 ms. However, without using caching the time required to fetch location information can take up to 193 ms. When the caching amount increases, the time needed to fetch emulated data decreases at 3 seconds of cache, we could reach 57 ms which is very close the native 55 ms. However, increasing the cache amount to 4 seconds results in 61.3 ms as shown in Figure 4.12 worsen the time required to fetch the location, this is due to substantial data transfer and SQL processing time.

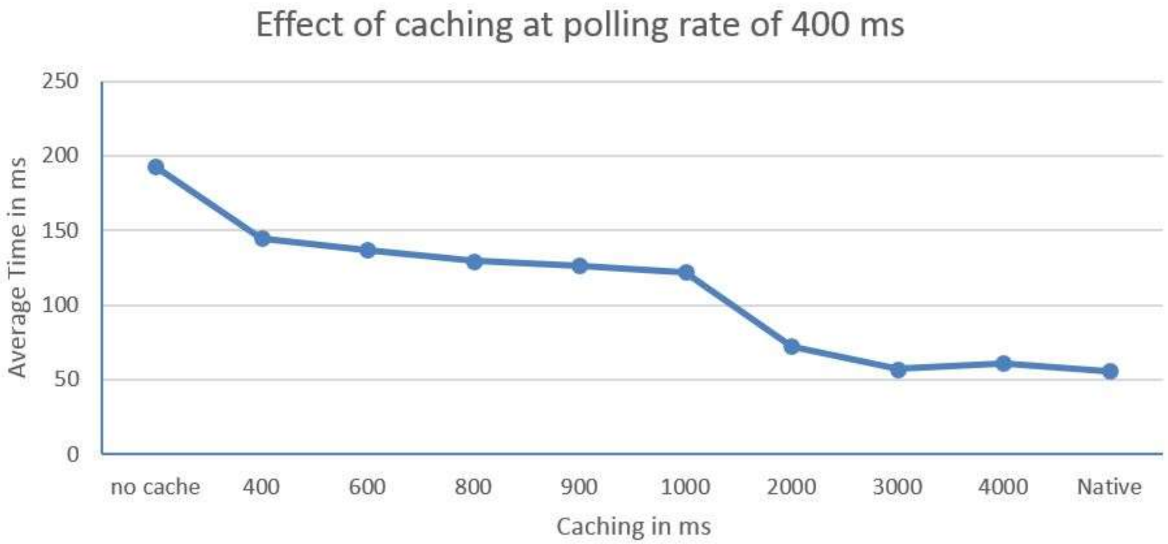


Figure 4.12: Effect of caching on average retrieval time

We study the effect of the location request interval on the success cache hit, as shown in Figure 4.13.When we are using a cache of 600 ms and the application is requesting data each 200 ms, the probability of cache success is about 75% and goes to zero when the request interval is more 1 second or more. This affects the time required to fetch the data as shown in Figure 4.13 it can take up to 275 ms to fetch location with 1 second of application request interval and 600 ms of cache.

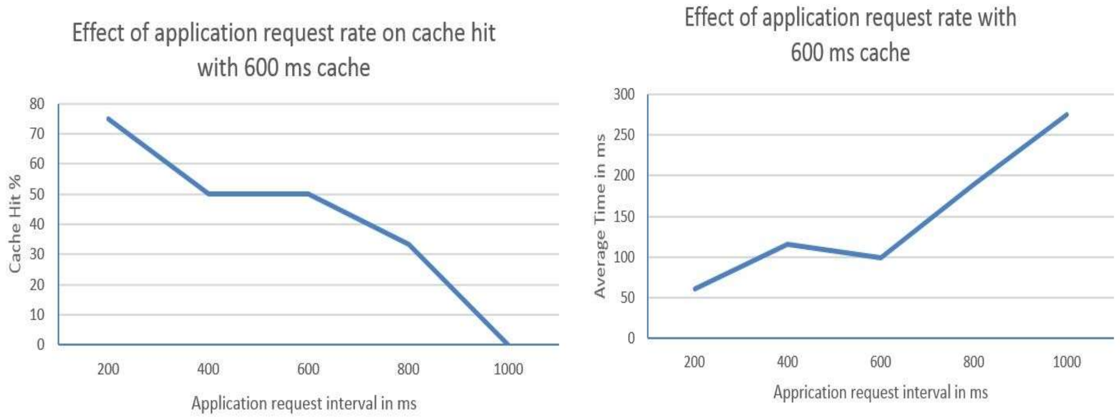


Figure 4.13: Effect of request interval time on cache hit percentage (left) and average time

4.6 Conclusion

VANET applications emerge as a promising market for both safety and infotainment application development. However, there is a need for a realistic yet flexible testbed that is capable of emulating VANET environments, enabling user testing and evaluating QoE. In this paper, we studied the requirements of such a testbed and introduced a flexible testbed that is tailored for VANET applications' needs.

To ensure the maximum usability of the testbed, we created and Android O.S. based testbed that can be deployed on virtual machines, real hardware and emulators. The proposed testbed does not only emulate location and mobility for VANET but also emulates wireless channel qualities and enables wireless P2P communications. To our knowledge, no available testbed is capable of testing applications on VANET environments with wireless emulation capabilities.

We adopt multistage emulation, where location, wireless properties, and all other sensor data are processed first, then converted to the testbed's database format, and finally, they are consumed by testbed's nodes. The proposed testbed is not coupled to specific emulators or simulators.

Our results show the suitability of our testbed for the location and wireless emulation requirements of VANET. Moreover, the introduced caching entity is capable of reducing the time of fetching emulated data to be similar to native one.

CHAPTER 5 ARTICLE 4: A COOPERATIVE ROAD TOPOLOGY BASED HANDOFF MANAGEMENT SCHEME

Hazem Ahmed
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
hazem.ahmed@polymtl.ca

Samuel Pierre
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
samuel.pierre@polymtl.ca

Alejandro Quintero
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
alejandro.quintero@polymtl.ca

Submitted to IEEE Transactions of Vehicular Technologies, 2016.

ABSTRACT

Selecting the most stable partner on Vehicular Ad-Hoc Network (VANET) is a challenging task due to its unique mobility, driver behavior, and networking requirements. Although new partner selection protocols are introduced for MANET and general mobility, there is a lack of a partner selection protocol that is suitable for VANET. This paper discusses the requirements of such

protocol and presents a handoff protocol that is tailored to VANET requirements. We leverage the use of partner assisted handoff by introducing Vehicle Link Expiration Time (VLET) metric that is designed to maximize inter-vehicular connections. The suitability of the proposed protocol is carefully studied; then compared to the state-of-the-art protocols. In our performance analysis, we compare average connection time, connection stability, and handoff delay and success rate. The results show that the proposed protocol enhances the handoff on VANET and can increase the average connection time among vehicles by about 200%.

5.1 Introduction

The vehicle industry is one of the pillars of North America`s economy. In 2014, more than 1.8 million[82] new vehicles are sold via Canada for a total price of more than $59.99 billion[83]. In the recent years, there has been a significant jump in vehicular technologies; companies are competing to provide more safety and entertainment applications and services to their customers. The vehicular network safety and infotainment industry is a relatively new and promising area of business.

With the latest dramatic expansion in the vehicle electronics and wireless technology, advanced wireless applications are developed to suit the vehicular environment. Such applications include both safety and infotainment applications. Infotainment applications include Video-On-Demand, Voice over IP (VoIP), live streaming, news and weather broadcast, Internet Protocol Television (IPTV) and interactive games. Safety applications include smart lane change, road hazard warning, accident notifications, and cooperative collision avoidance. Along with the rapid growth of the Internet and the continuous efforts of cellular communication networks to provide mobility support, more applications are no longer restricted to wired networks and can run consistently on wireless networks. The cornerstone of cooperative applications is partner node selection.

VANET has unique characteristics that differ it from Mobile Ad hoc Networks (MANET). VANET is based on vehicles that provide large physical structures of vehicles and availability of rich power source. This enables vehicles of hosting powerful equipment without worrying about power consumption and the size of equipment. Besides, vehicles in VANET do not move randomly. However, they move on predefined roads and streets with relatively high speed.

VANETs utilize wireless communication among vehicles to facilitate applications that are cooperative and distributed in nature as well as connecting to existing network infrastructure to facilitate and consume services. However, VANETs suffer from several Quality of Service (QoS) problems. Handoff is one of the major challenges that affect the QoS on VANET.

The original Internet Protocol (IP) is not suitable for mobile nodes [84]. The need for an enhanced IP extension that is appropriate for the mobile node is the main force behind the introduction of Mobile Internet Protocol (MIP)[84]. By introducing IPV6 protocol, the MIP evolves to MIPv6. The MIPv6 add the mobility support to IP protocol by permitting Mobile Nodes (MN) to move through different domains. The problem of large mobile network handoff delay caused by Binding Update(BU) messages sent to, the usually far, Home Agent (HA) is the driver behind the introduction of Hierarchical Mobile IPv6 (HMIPv6) [85]. Conversely, HMIPv6 fails to solve the other MIPv6 inherited QoS problems especially the long delays in handoff and packet loss during handoffs. Fast Handover for Mobile IPv6 (FMIPv6) [86], addresses the handoff latency of MIPv6 and presents an efficient solution to it. Before the actual roaming to a new network domain, MN detects the new network using link layer information. The FMIPv6 addresses the packet loss problem while in roaming by tunneling all packets forwarded to PAR to NAR and buffering them in NAR. FMIPv6 theoretically prevents any packet loss due to roaming to the expected NAR. Theoretically, only in case of unexpected NAR buffer overflow, there will be a packet loss. The FMIPv6 improves the QoS significantly compared to MIPv6 and HMIPv6, however; it inherits the same delay problems from MIPv6 that were solved by HMIPv6.

To handle all the mobility concerns by the network only seamlessly to MN, Gundavelli et al. in[87], propose Proxy Mobile IPv6 (PMIPv6). PMIPv6 simplifies the design of MN IP signaling and interaction. Besides, since the network side handles the mobility, it is easy to integrate multiple access protocols seamlessly to moving vehicles. However, PMIPv6 does not consider any specific vehicular mobility and suffers from packet loss and handoff delay similar to MIPv6 and HMIPv6.

To reduce handoff signaling and improve handoff delay, Devarapalli et al. in [88], propose a standard for adding network mobility support to IPv6 namely: Network Mobility (NEMO) Basic Support (NEMO-BS). The protocol groups all the nodes that move together (e.g. multiple IPv6 nodes in the same train or bus) to be handled with a Mobility Router (MR) as one entity. All mobility interactions are done between the MR and the Access Router (AR) while no nodes behind the MR are handling any mobility-related tasks. The protocol uses the same CoA and HA concepts to handle mobility at the network side. Due to the ease of implementation and the fact that all the mobility actions are dealt with by the network side, there is a broad adoption of NEMO-BS protocol with different router implementation and support. However, NEMO-BS is not suitable for VANETs and still suffering from high latency and hand off problems[89].

Cespedes et al. in[89], focus on the application of NEMO-BS on VANETs. The authors show that the lack of Route Optimization (RO) functionality in the original protocol causes several performance degradations. The authors compare the state of the art research related to the enhancements of NEMO-BS to support RO. This research is one of few types of research that discuss the NEMO-BS challenges and solutions. The authors mentioned that there is a need for a new routing and inter-vehicular protocols that depend on the geographic location but not necessary the currently used greedy forwarding technique. Also, they show a need to improve the handoff QoS to the current NEMO-BS. We are solving this issue by selecting the best partner based on the network topology.

Baldessari et al. in[90], provide a solution to integrate geographic routing in C2C-CC architecture to NEMO-BS to enable the application of NEMO-BS on the VANET environment. The proposed solution uses the geographical routing as a sub-IPv6 layer, which hides the implementation from the IPv6 avoiding IPv6 incompatibility. The authors show that using that adding a new tunnel between MR and AR in addition to the standard IPv6 tunnel between MR and HA is enhancing the QoS and reducing the total network signaling.

In Partner assisted handoff mechanism HMIPv6 (P_HMIPv6)[16] Chen et al., propose a technique to reduce the handoff delay. The P_HIMPv6 uses one of the MNs in the future MAP domain to work as a Partner Node (PN), which is similar to IEEE 802.16J RS[16] and helps the MN to pre-perform the unfinished work of the L3 handoff. To select a PN by MN, PN should be static and located near to the MN. The localization proximity is measured using Received Signal Strength Indicator (RSSI) to decide if the relay node is near or not. The P_HIMPv6 is proven using mathematical analysis and simulation to reduce the handoff delay and improve the QoS. The methodology in P_HMIPv6 is similar to the FHMIPv6 in the sense that the L3 handling is performed before the actual roaming to the new region. The advantage of the P_HMIPv6 is that it does not require special router buffering or hardware upgrades since it using MNs as PSs.

Using RSSI to select a relay node or PN is not accurate in vehicular networks, simply because the road topology along with direction and speed of the PN have a substantial effect on the connectivity between the MN and PN. To address these issues, we propose a Vehicular Link Expiration Time (VLET) as a parameter to select the best PN. The proposed protocol is making use of static road topology along with the future route of MN and the route probabilities of PNs to choose the best PN that will be connected MN the longest possible. Extensive simulation and mathematical modeling are conducted to evaluate the new protocol and to compare it with the original P_HMIPv6 and the state of the art protocols.

5.2 Problem Analysis

In this section, we analyze the handoff problem in VANET and challenges of selecting an optimum PN that will be in the range of MN for a sufficient period to complete the handoff process. Firstly we will discuss problems with a handoff on VANET, and then we discuss the popular RSSI and SNR as a metric to select best PN. Finally, we discuss VANET scenario where the state of the art partner selection for handoff algorithm fail.

5.2.1 VANET Handoff

Providing seamless mobility is a critical requirement for the next network generation applications[16, 91]. Seamless Handoff QoS assurance during handoff is a very challenging issue, since roaming through several networks triggers handoff between homogeneous and heterogeneous networks[16]. Handoff in MIP environment involves L2 and L3 delays, packet loss, application layer negotiation, and special hardware resources to facilitate the handoff [16, 92-94].

The handoff problem is amplified in VANET due to high mobility and the unpredictability of traffic, road conditions, and driver behavior. High mobility increases the number of networks that vehicles roam through, increases the number of handoff requests on VANET and decreases the handoff success rate. Moreover, high mobility adds more pressure on the vehicles to complete the handoff process in shorter time.

There is an existing problem with the proactive handoff protocols where MN does not have enough time to finish the handoff process[16]. High mobility of vehicles is adding more pressure on vehicle handoff protocols to handle requests in short time. Furthermore, the unpredictability of road traffic conditions and driver behavior contribute to variable vehicle network density which leads to network interference[95] and a performance bottleneck in densely connected downtown scenarios or improperly well-connected clustered silos in the case of highway scenarios. There is a need for a better QoS based handoff scheme to for VANET.

5.2.2 RSSI Based Protocols

RSSI and SNR are one of the first methods used to select a partner node. The main advantage of using this approach is the ability to measure RSSI and selection of a PN with no need for extra information to be shared and using information available on the wireless interface.

Chen et al. in [17], propose a relay assisted handover in Cognitive LTE Networks. A partner node is used as a relay to help MNs to connect to their previous and next BSs using available spectrum slots. The results show that the network throughput is increased, and the end to end delay is reduced significantly. The discovery phase of PN is done by periodically sensing the environment and track the PN that is in statically stays in range and by evaluating Signal to Noise Ratio (SNR). However, using SNR alone to select a PN is not suitable for VANET, where a vehicle is maybe driving in the opposite direction but close to the AP as discussed in [18]. Bhaskar et al. in [96], propose Noise Resilient Reduced Registration Time Care-of Mobile IP (NR-RRTC) protocol and study the drawbacks of using RSSI in 4G MIP networks. The main contribution of this work is assuming that the signal is degrading when an MN moves towards the boundary limits as opposed to the original protocol that considers the signal is uniform throughout the boundary area.

Using a PN to help in the handoff process is proven to enhance the handoff QoS [17], [18], [16]. However, the proposed optimum PN selection protocols in the literature are not suitable for VANET. Using RSSI is proven to result is selecting a wrong partner in MANET [18]. Since the fact that if a PN is near to an MN is not granting they will last in range until the handoff process is completed, mainly due to two reasons: the first is selecting a nearby PN that is moving in an opposite direction with a fast speed, as shown in Fig.5.1 the PN will be soon out of range. The second is selecting a PN which moves with much faster speed than the MN; this results in both MN and PN will not be in range soon. However, there might be other PNs in the range of MN that are moving almost with the same speed and will stay in range longer and got discarded. Merely because they are not the nearest to the MN.

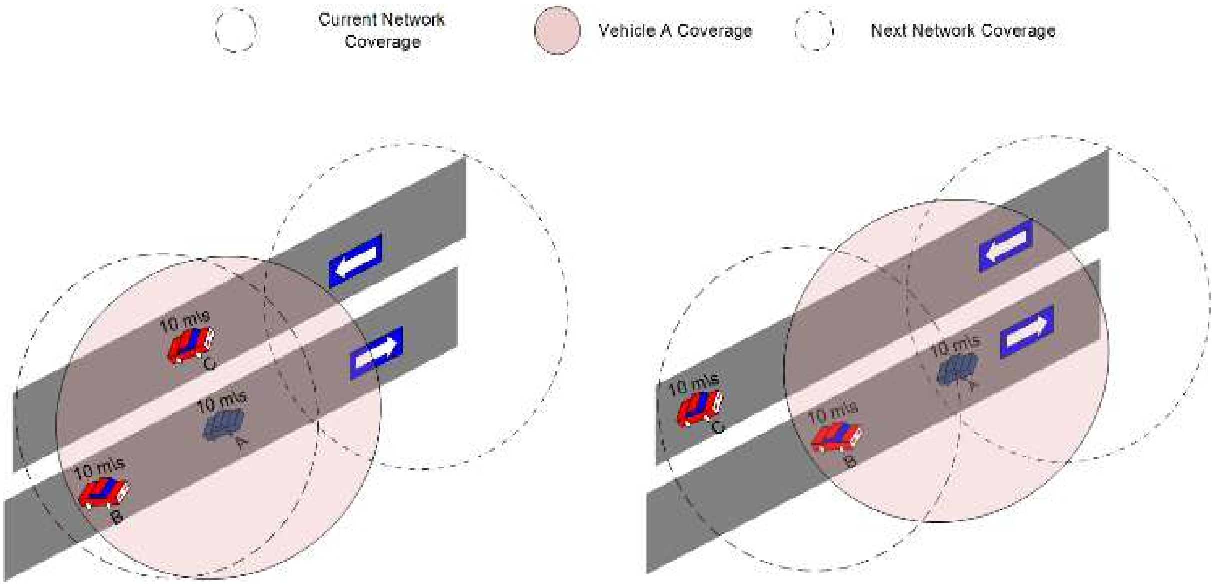


Figure 5.1 : RSSI PN selection scenario. In (a) MN "A" is selecting a PN "C" based on RSSI. In (b) after some time elapsed, PN "C" leaves the communication range of "A" while "B" stays in range.

5.2.3 Smart Partner Selection Protocols

Selection of PN based on RSSI only is not valid for most of the MANET scenarios [18], where PNs close to MN are not necessarily the best partner that will stay longer in range. Hence, more information and smart algorithms are required to select the best partner to maximize inter-vehicular connectivity.

Sharing location and destination of a PN is one of the first successful attempts to avoid using RSSI. Choi et al. in [97], provide integration between geographical routing to the Car-to-Car Communication Consortium (C2C-CC) to support multicasting. This defines a virtual point-to-point link by changing the IPv6 standard protocol to be location aware to enable the multicast capabilities without depending on the link layer. Position based greedy forwarding is used to implement the multicast function. However, this solution does not deal with IP mobility and handoff issues and modify the original IPv6 standard, which may cause interoperability problems in the future.

Similarly, location and destination are used to enhance PMIPv6, Sandonis et al. in[9], exhibit an integration framework between geo-networking and PMIPv6. The authors integrate the single hop

PMIPv6 to the European architecture standard to provide a multi-hop adaptation. The research proposes several scenarios using real mobility traces to evaluate the performance of Internet connectivity on VANET. However, in this research, the Mobility Access Gateway (MAG) is always stationary, and the case where the MAG is mobile is not addressed.

Using location and destination in a geo-networking like environment is better than using RSSI, however, there is a security threat of sharing destination with other nodes, there should be a maintenance to destination and locations of peers, and finally it can be expensive to share location and destination information among nodes.

One of the major improvements in a smart selection of a PN is provided by Taleb et al. in[18]. The authors introduce using Link Expiration Time (LET)[98] to select the best PN to assist in the handoff and to overcome the problems of using RSSI only in P_HMIPv6. LET uses relative position, speed, and direction of a PN to an MN to predict the connection period between them. The authors used the Exponential Moving Average (EMA) to introduce Connection Stability Aware P_HMIPv6 (CSA_P_HMIPv6), where the EMA is employed to estimate the duration of connections with each expected PN to select the most stable connection among them.

Despite not using RSSI indicator for selecting a PN and using LET instead improved the performance, we show that LET is not suitable for VANETs where vehicles are using routes as opposed to the random or directed mobility considered in the MANET scenario.

Although the use of LET as a metric to select the best PN eliminates the mentioned problems of RSSI in MANET, it is not suitable for VANET environment since PNs do not always continue in their current direction. For example, if both the MN and PN are moving in the same direction with the same speed, according to LET calculations they will be in range theoretically forever. Nevertheless, it fails to handle simple vehicle scenarios. Fig. 5.2 shows example scenarios that LET fails to select the best partner in range. In all the examples we will consider vehicle "A" to be the moving MN that is looking for a PN to complete the

handoff.

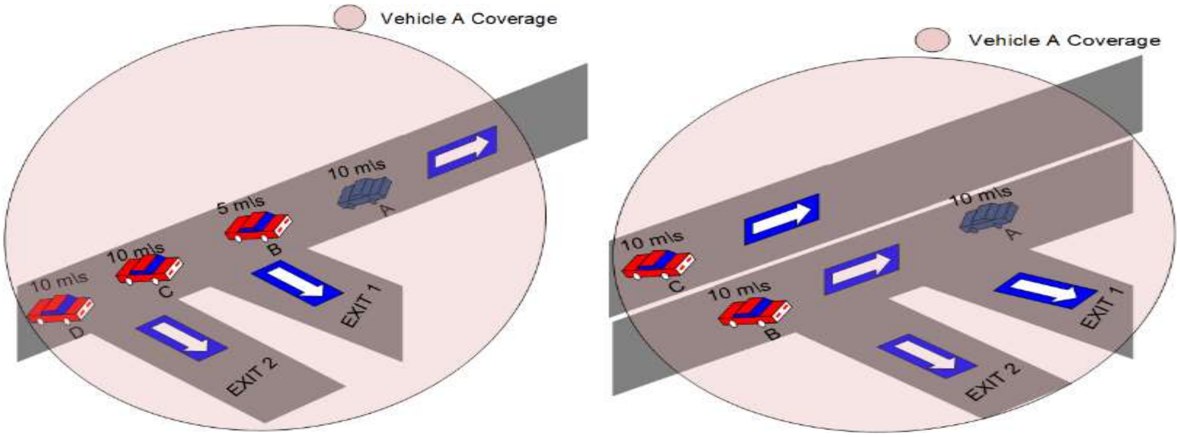
The simplest vehicular scenario is to consider all vehicles are sharing the same road segment and driving in the same direction. In this case, the LET algorithm may falsely select a PN that is cruising at a similar speed, but it may leave the road segment sooner than a different vehicle that has no other option but to continues on the same road segment. In Fig. 5.2. (a), vehicle "A" will select vehicle “C” as the first choice for optimum PN using LET and “D” as a second one; since they are cruising at the same speed. However, both “D” and “C” has an option to leave the road segment which will result in shorter connection time. Although vehicle “B” is cruising at a slower speed, it has no other option but following “A.”

Selecting a PN that shares the road segment is not always the most optimum choice as shown in Fig. 5.2. (b), where both vehicles “A” and “B” are sharing the same road. However, since vehicle “B” may exit from “EXIT 1” or “EXIT 2”, vehicle “C” will work as a better PN for “A” since it does not have a chance to leave the road, and they will stay longer in range.

The future route of the MN has a significant effect on selecting the correct PN, in Fig. 5.2 (c) vehicle “A” will take “EXIT 1”, in this case, “B” will serve as a better PN as compared to “C”. Sharing routes between vehicles could be a solution to this problem. However, it has major security drawbacks. Moreover, routes of PNs are not static and can be changed unpredictably based on the driver’s decision. This requires updating routes among PNs which decreases the throughput.

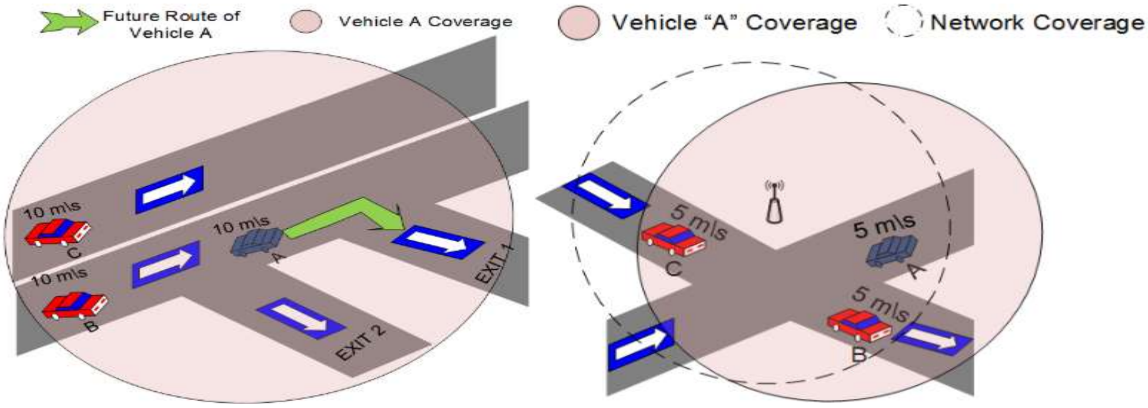
Finally, most partner selection handoff protocols in the literature fail to select the right partner in a situation like the one described in Fig. 5.2. (d). Where both RSSI and LET based methodologies will select “B” as a PN for “A”, yet in fact, “C” is a better partner. Because “B” will be leaving the road segment while “C” will be joining and is expected to stay in range of “A” for a longer time.

For all these reasons, there is a need for a partner selection protocol that tailored for VANET needs. In the next section, we describe our proposed protocol that takes into account VANET characteristics without demanding extra information be communicated among vehicles.



(a) MN and PNs are sharing the same road segment

(b) MN and PNs are on different segments



(c) MN will leave current road segment

(d) leaving and joining road segment

Figure 5.2: Several scenarios where state of the art partner selection algorithms fails to select the right partner.

5.3 Proposed Partner Selection Protocol

In this section, we will introduce enhancements for the P_HMIPv6 to be suitable for the vehicular environment. Firstly, we describe the proposed partner selection protocol, and then we introduce our Vehicular Link Expiration Time (VLET) that can select an optimum PN for an MN in the vehicular environment, and then we formulate the mathematical model behind. Furthermore, we introduce an enhancement on the original protocol to increase the connectivity and robustness of P_HMIPv6 that aims at increasing the connectivity while in handoff.

5.3.1 A VANET Tailored Partner Selection Protocol

The proposed algorithm aims at selecting the best PN that will stay in range of the MN for the largest period, taking into account the VANET topology. The protocol assumes the availability of a digital map system and Global Positioning System (GPS). With the recent dramatic reduction in the cost of the onboard vehicle GPS systems, these assumptions are realistic. In fact, the new V2V standards assume the availability of a GPS location and broadcast this location for safety applications purposes [99]. Moreover, live traffic information is available on almost all new GPS systems using Traffic Message Channel (TMC); thus, our protocol makes use of this feature as well to improve the best PN selection.

The VLET protocol does not require PNs to share their expected routes with the roaming MN, but rather it uses their current location, GPS device and a local digital map system to predict neighbors' routes in the future. The proposed protocol reduces the messaging cost and delay resulted by frequent location message update pull due to partner's disconnections in the middle of the handoff process. Additionally, in VANET critical handoff situations there are isn't enough time to select a different PN if the initial selection failed, this makes the initial choice of a PN essential for handoff operation.

The implementation of such protocol is practical using a database such as MongoDB [100], which supports natively geospatial data transactions. It takes the database engine to search and retrieve all road information data in the range of 5 KM proximity of a given GPS coordinates about 0.09 ms [101].

5.3.2 Proposed Partner Selection Algorithm

In this section, we provide an overview of the proposed partner selection protocol using VLET metric.

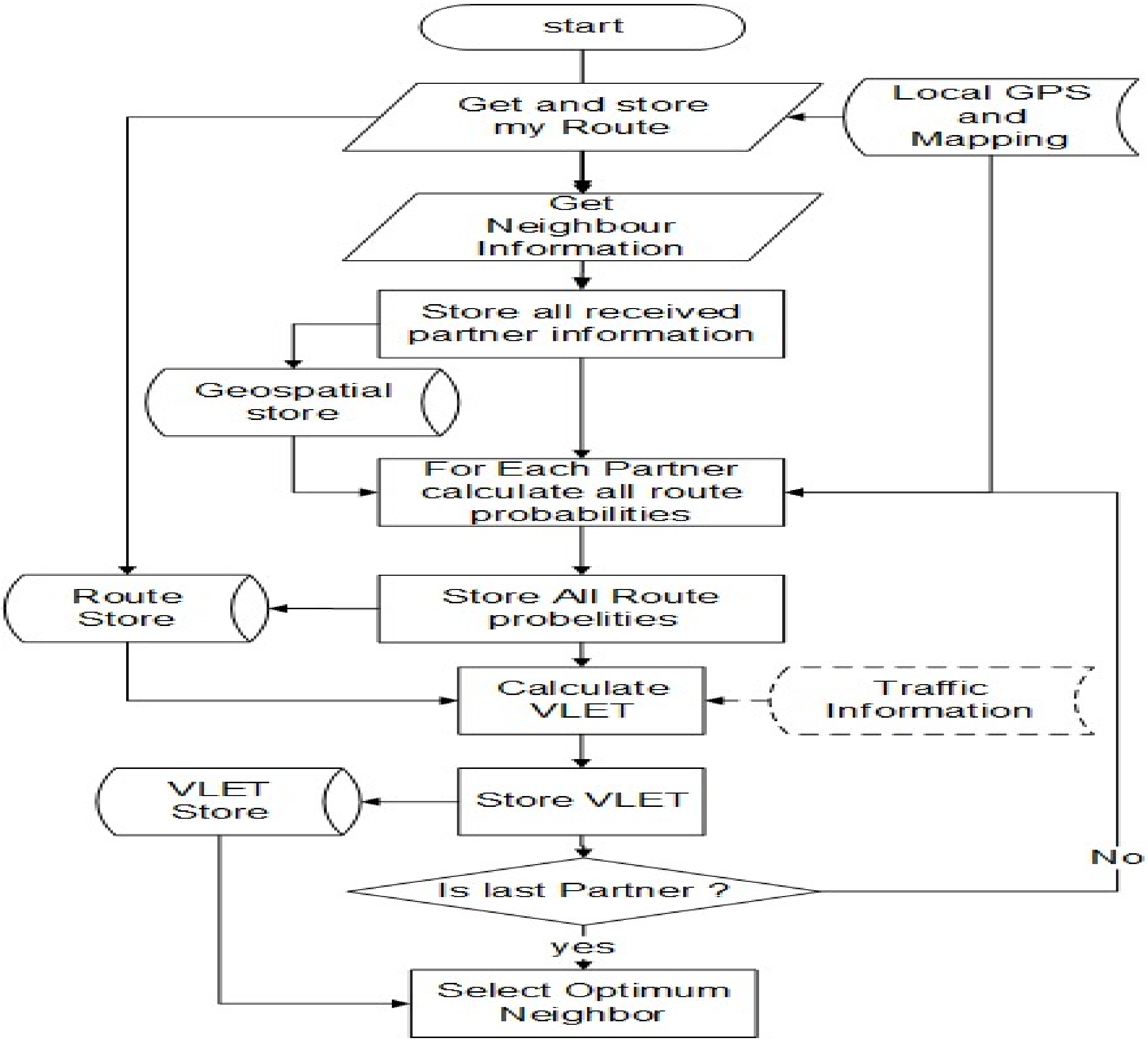


Figure 5.3 : The Proposed Partner Selection Protocol

As shown in Fig. 5.3, once MN decides to start the handoff process, it retrieves the current route from the onboard GPS system, and then the MN sends a broadcast a message to all PNs in its direct vicinity and within its transmission range, all NPs should reply immediately with their speed, position, and heading. We have to note that any PN does not retransmit location request message. The MN waits for a specific timeout relative to the wireless environment to receive all partners' replies. In the case of DSRC, the location of partners can be detected from the beacon safety messages. The VLET algorithm then saves partners' information to a geospatial store.

By using the local digital map system, we estimate all the route probabilities for each partner. For example, in Fig. 5.2. (d) the route probabilities for vehicle "B" is only one to move straight, but for vehicle “C” there are two probabilities the first is to go straight and then take Exit 1, and the second is to continue straight and does not take the exit. However for vehicle “D” there are three probabilities, the first is to take Exit 2, the second is to take Exit 1, and the third is to continue straight in the direction of vehicle “A”.

To reduce the number of possible routes for each PN, the MN is using the onboard digital map to infer the future road segment speed along with the current relative speed of the PN to limit the number of future routes for only the period of the expected handoff. For simplicity, a constant threshold (e.g. five sec.) can be used, where rendezvous time between MN and PN will not exceed this threshold.

Once all the required routes calculated for a PN, it is stored and then used to calculate the VLET. VLET calculates the rendezvous time between for each segment of the future route of the MN and each segment of the PN’s future calculated routes, and then an optimization function is used to get the most suitable expected link expiry time. The VLET algorithm is using live traffic information to enhance the future segment and route speed. For example, if in Fig. 5.2. (d) the road segment speed of Exit 1 as retrieved from the local digital map is 10 m/s but the live traffic information indicates that the Exit 1 road is slightly congested, and average speeds are 5 m/s then the algorithm will use the actual value to enhance the link expiration estimation.

Once VLET is calculated for all partners, we select the PN with the largest VLET which means it the most likely to say in the MN’s range of communication for the longest period possible under the current road conditions. In case that the calculated VLET is not sufficient for performing handoff, we prepare a list where MN will switch to the second greatest VLET time once the first one is expired.

In Fig. 5.4, vehicle “A” is taking the “EXIT 1”. Then using the VLET protocol, for both partners in the range all the possible routes will be calculated. For vehicle C there are two possibilities, straight forward “C2” or taking the exit “C1” and for vehicle “B” there will be three possibilities: “B1” where vehicle “B” will follow vehicle “A” and take “EXIT 1”, “B2” where vehicle “B” will go straight forward and not taking “EXIT 1”, or “B3” where vehicle “B” will take exit “EXIT 2”. VLET will select vehicle “B” instead of “C” since there is a higher probability of staying in range.

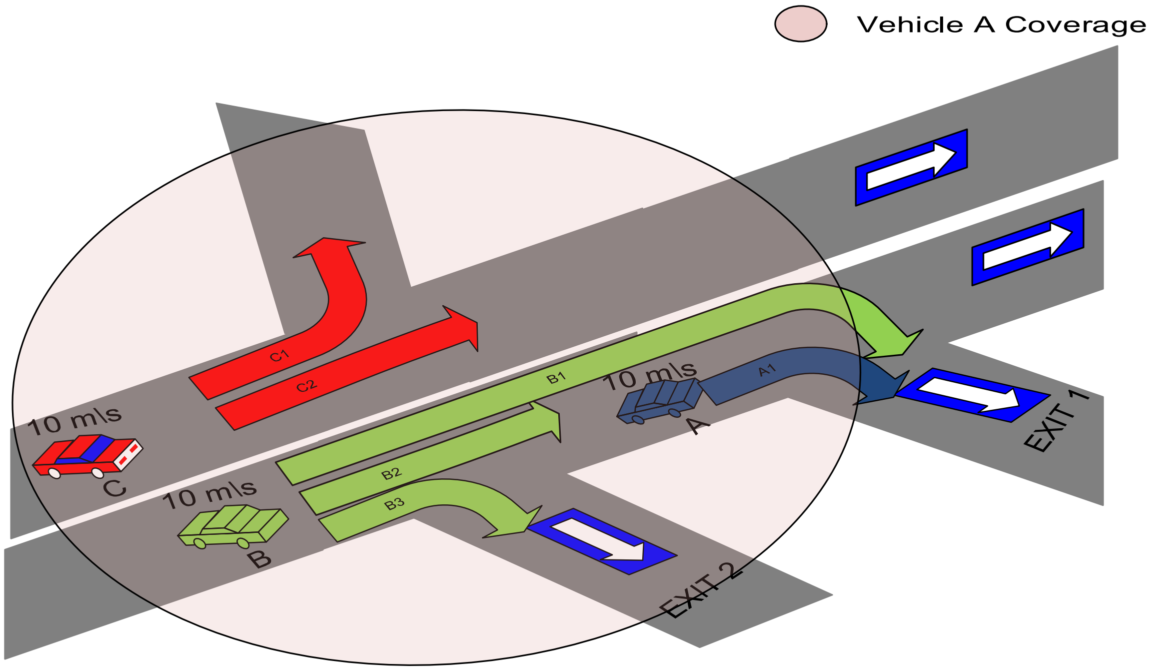


Figure 5.4: A simple scenario of using road topology to select the best partner using the expected routes.

5.3.3 Mathematical Formulation of VLET

In this section, we show in detail how the VLET metric is calculated. As discussed in the previous section, VLET maximizes the connection time between a roaming vehicle S and a partner vehicle P. Since the future route of S is already known by using the onboard GPS system; there is no need to assume different routes. However, we assume that the path of S will contain $NS\_Segment$ segments, these road segments are already provided by the digital maps of the GPS.

Since we know only the current position of P and not the planned route, then we will use all the possible routes (NP_{routes}) for P in range of S. Similar to S each route j will be divided to number of segments $NP_{Segment_j}$.

The number of expected routes of P can be calculated as the direct route from P to S, for the particular case that P will follow S, in addition to all route possibilities between P and S. For example, if there is only one exit between P and S, then P will have two possible routes one route is to take the exit and the other is the direct one.

Assume that R^a denotes the expected route a and R_b^a denotes route's segment b for route a.

Then the position of the P at a particular time T on route R^j is donated as $R^j|_T$. Segment Expiration Time at segment i (SET_i) is defined as the minimum LET between S and P at the time S is at the end of the segment i for each segment in the future route of S. Also, there is a probability that P will take any of the expected routes, therefore we assign a priority β_j that corresponds to the probability that vehicle P will take route j and $\sum_{j=1}^{NP_{routes}} \beta_j = 1$.

Then $\mathrm{SET_i}$ can be calculated as shown in Eq. (1).

$$\mathrm{SET_i}\left(\mathrm{S}|_{\mathrm{Ts_i}}, \mathrm{P}\right) = \mathrm{MIN}\left(\beta_1 \ \mathrm{LET}\left(\mathrm{S}|_{\mathrm{Ts_i}}, R^1|_{\mathrm{Ts_i}}\right), \ldots.., \beta_\mathrm{j} \ \mathrm{LET}\left(\mathrm{S}|_{\mathrm{Ts_i}}, R^j|_{\mathrm{Ts_i}}\right)\right) \qquad (1)$$

Where Ts_i is the expected time S will reach the end of segment i, $S|_{Ts_i}$ is the expected location of S a time Ts_i and $R^j|_{\mathrm{Ts_i}}$ is the expected location of P at the time Ts_i on route j. LET between S and P can be defined the same as described in [98] and shown in eq.(2)

$$\mathrm{LET(S, P)} = \frac{-(\mathrm{ab+cd})+\sqrt{(\mathrm{a^2+c^2})\mathrm{r^2}-(\mathrm{ad-bc})^2}}{\mathrm{a^2+c^2}} \qquad (2)$$

Where $a = v_s \cos\theta_s - v_p \cos\theta_p$, $b = x_s - x_p$, $c = v_s \sin\theta_s - v_p \sin\theta_p$ and $d = y_s - y_p$, $(v_s, \theta_s, x_s, y_s)$ are the speed, angle of direction, location horizontal component, and location vertical component of S and similarly for P $(v_p, \theta_p, x_p, y_p)$.

Each route of P may contain several segments, so location of P is located on route segment k ($\mathrm{R_k^j}$), then $R^j|_{\mathrm{Ts_i}}$ can be calculated as $R^j|_{\mathrm{Ts_i}} = R_k^j|_{\mathrm{Ts_{i-1} \le Tp_k \le Ts_i}}$ where $\mathrm{Tp_k}$ is the expected time P will reach the end of segment k.

To calculate Ts_i and Tp_k we assume α_s, α_p as the velocity relaxation coefficient for S and P respectively. Then $\alpha_\mathrm{S} = \frac{V_c^s}{V_{0\_avg}^s}$ and $\alpha_\mathrm{p} = \frac{V_c^p}{V_{0\_avg}^p}$ where V_c^s, V_c^p are the current velocities for S and P respectively and $V_{0\_avg}^s$ and $V_{0\_avg}^p$ are the velocities of the average segment's speeds for S and P respectively. The average segment speed can be obtained using the updated traffic information with the usage of Traffic message Channel (TMC) or in case of the unavailability of TMC the segment speed limit is used. The proposed velocity relaxation $\alpha_\mathrm{S}, \alpha_\mathrm{p}$ are used to enhance the prediction of the moving speed of S and P. For example, if vehicle P is cursing at speed of 4 m/s but the average speed is 3 m/s for the current road segment, then we assume the that P may cruise with a speed ratio of 4/3 of the average speed other sophisticated relaxation equations such as exponential can be used to enhance the future speed prediction. Then Ts_i and Tp_k are calculated as shown in eq. (3).

$$Ts_i = \alpha_S \frac{V^S_{i\_avg}}{D^S_i} + Ts_{i-1} \; and \; Tp_k = \alpha_p \frac{V^p_{k\_avg}}{D^p_k} + Tp_{k-1} \qquad (3)$$

We used α_p and α_S to estimate a more accurate future estimation for partner and source vehicles respectively. These coefficient can be modeled using a more complex relaxation algorithms however, we assumed that they will be the same for simplicity reasons.

Finally, VLET can be obtained as shown in Eq. (4)

$$VLET\ (S,P) = \sum_{i=1}^{\mathrm{NS\_Segment}} \mathrm{SET_i}\left(S|_{Ts_i}, P\right) \qquad (4)$$

Both RSSI and LET based algorithms are not VANET topology aware. Hence, they fail to predict route probabilities. As mentioned earlier, RSSI based methodologies will fail to select vehicles moving in the same direction which are essential for prolonging the link between the two vehicles.

The original LET algorithm that is used in [18] assumes that both P and S will continue on the initial speed and direction, that means there is only one expected route which contains only one segment, which represents the worst case scenario of failure of routes prediction:

$$NP_{routes} = 1, NS_{Segment} = 1, NP_{Segment} = 1, N_{P_{routes}} = 1,$$

$$VLET(S,P) = LET(S,P) \qquad (5)$$

5.4 **Performance Analysis and Results**

In this section, we study in depth the performance of the proposed protocol. We focus on the enhancements and the stability of the proposed protocol, where the main variables are the transmission range and the network delay (mainly DAD). First, we discuss our simulation setup, and then we show performance result for connection time, handoff success rate, throughput and finally handoff cost.

5.4.1 Simulation Setup

We pay specific attention to the realistic vehicular environment. Therefore, we used a real network topology subset that is generated from OpenStreetMaps[102] with traffic lights, turns, exits, and U-turns. Then we convert the maps to a format that is suitable for a realistic urban simulator SUMO for simulating the mobility.

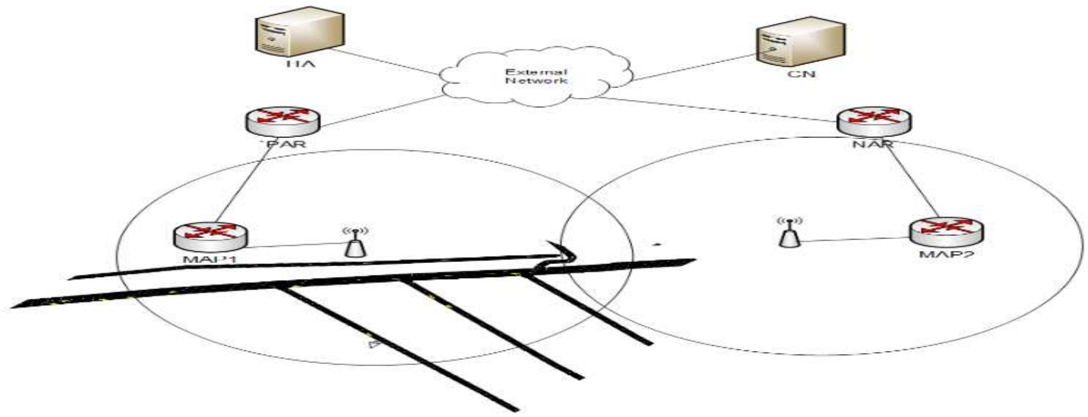


Figure 5.5: Network Simulation Model

We use a two-way street where one way does not contain any exits, and the other contains three exists. We are interested in studying the vehicles that are moving in the direction where there are exists. There are 50 vehicles in our simulation, 26 each direction. Vehicle routes are generated randomly where each vehicle can select to take one of the three exits or to continue straight forward. SUMO[40] simulates lane changes, acceleration patterns, and driver behavior. Also, it handles the road configuration regarding speed, direction, traffic signs, and turns. We developed a program to extract all topology results from the SUMO simulation at each step for easier results manipulation. These results include each vehicle and its direct neighbors, the distance between neighbors, and their speed. The generated mobility trace file is converted and used in NS2[103] to analyze the network performance metrics.

The simulation setup assumes two adjacent networks the vehicles move from one domain to the second as shown in Fig. 5.5. As mentioned before the mobility is generated using SUMO and we used several speed classes. The considered network topology is along with topologies in [16] and [18] where the HA, CN, Next Access Router (NAR) and Previous Access Router (PAR) are connected through global network(e.g. Internet) and the propagation delay between AP and AR is set to 20 ms and the propagation delay between AR and MAP are set to 50 ms.

5.4.2 Vehicular Connection Time

Vehicular Connection time is a vital metric for any partner assisted handoff protocol to function. In the P_HMIPv6, a partner has to complete the handoff process using a partner. Otherwise, the handoff operation will be considered failed. In this section, we discuss the results of our proposed VLET and compare it to three partner selection techniques namely: RSSI, Direction and. LET. RSSI is the original technique used in P_HMIPv6, Direction is a protocol where we assume the partner vehicle shares its final destination with the roaming vehicle, and finally LET. We consider the average connection time between a vehicle and a partner throughout the simulation. If there is no partner in the direct vicinity, the interconnection time will be equal to zero. We run the simulation multiple times with different radiuses varies between 100 and 1,000 meters to accommodate different VANET wireless technologies.

The RSSI selection is almost not biased by the communication radius since the closest partner is always selected as shown in Fig. 5.6. However, when the communication radius is too small, this risks the unavailability of any vehicle in the range such as “V20” with 100 meters of communication radius in Fig. 5.6.

Figure 5.6: Communication radius effect on the average connection time using RSSI method.

Contrary to the RSSI, the Direction method is highly biased by the communication radius as shown in Fig. 5.7. The more communication radius, the worse the average connection time. This is due to the fact there are multiple vehicles are going to the same destination as the roaming vehicle. Also at the moment, they are going in the same direction. Selecting one of them randomly results in vehicles that may leave the communication range of the roaming one.

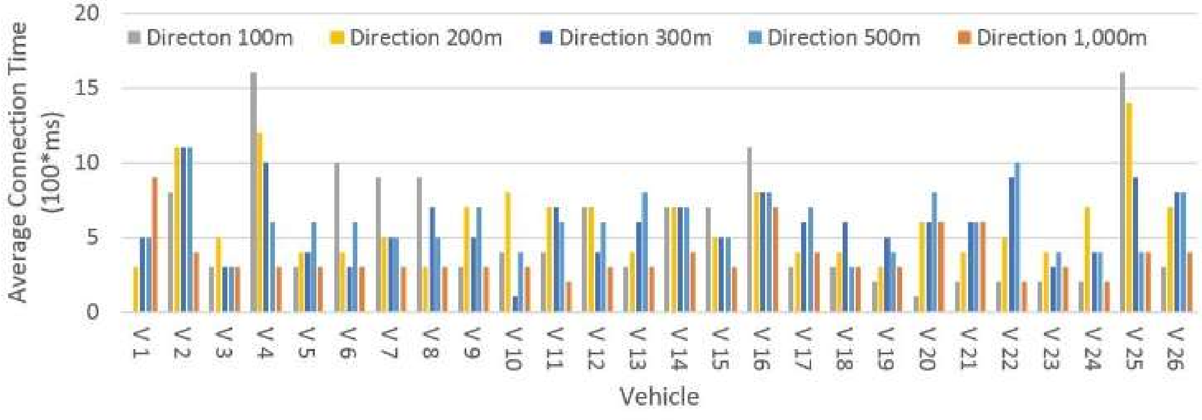


Figure 5.7: Communication radius effect on the average connection time using "Direction"

Contrary to the expectation that a bigger communication range will benefit LET selection algorithm, the results show that LET performs badly as shown in Fig. 5.8. The reason is when the range is greater; the LET tends to select vehicles that are far in the distance at the moment but driving in the same direction and with very similar speed. This causes the problems where PN vehicles may leave as discussed earlier.

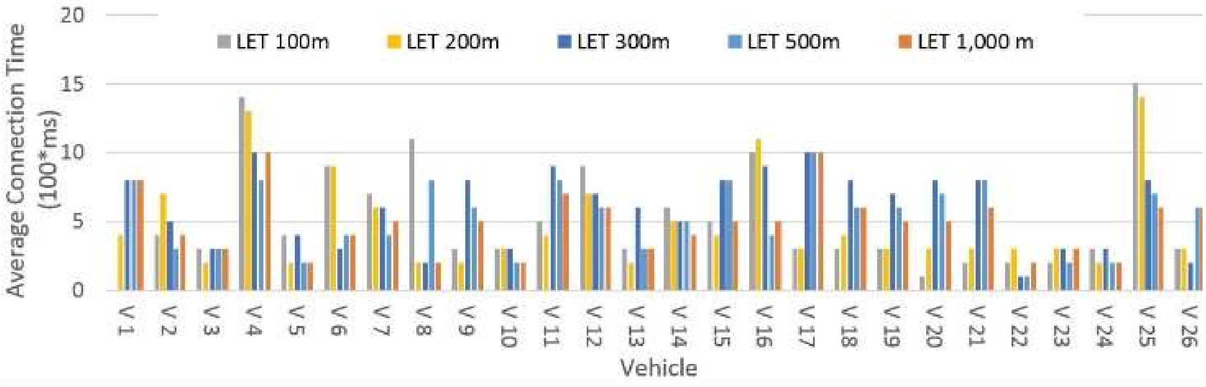


Figure 5.8: Communication radius effect on the average connection time using LET method.

Although both LET and VLET in bigger ranges can find a partner, it is critical to stay with the selected partner the longest possible to ensure the handoff success. VLET is topology aware. Therefore, the more radius provided, the more it selects a right partner as shown in Fig. 5.9.

In Fig. 5.9, we notice that VLET interconnection time is gradually enhancing with the range. More importantly, it is not getting worse by higher ranges. The VLET is stable for radiuses 300 m, 500 m and 1,000 m radiuses and the worst results are with 100 m radius where it is not easy to find partners in range.

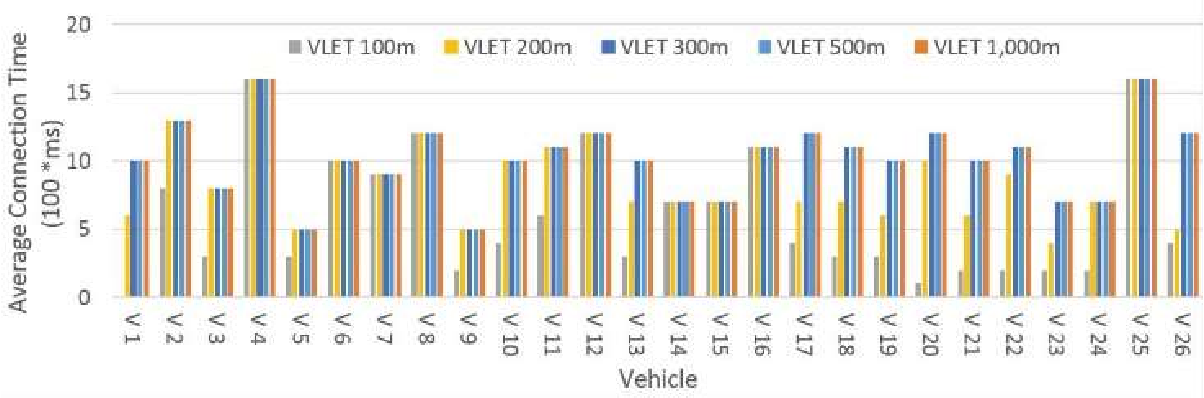


Figure 5.9: Communication radius effect on the average connection time using VLET method.

When the range is small, the selection is limited. In Fig 5.10, we compare the four methods with 100 m communication radius. We notice that the results are nearly similar except for the RSSI, which is much worse, this is due to the selection of opposite direction vehicle that goes out of range.

Figure 5.10: Comparison between VLET, LET, RSSI and Direction methods with small communication radius.

In Fig. 5.11, we compare the four methods at 500 m communication radius. We notice that VLET is superior as compared to all the others. LET is still better than both RSSI and Direction methods in most cases. However, since LET is not topology aware; it cannot compete with VLET. VLET is shown to enhance the average connection time by more than 175% as compared to the RSSI for a radius of 500m.

Figure 5.11: Comparison between VLET, LET, RSSI and Direction methods with 500 m communication radius.

5.4.3 Partner Handoff Success Rate

Handoff success rate is defined as the ability to complete the pre-handoff process using the first selected partner. As shown in Fig. 5.12, the success rate of all methods is about 100% when the radius is more than 200 m, and the pre-handoff delay is less than 150 ms. However, due to the shorter connection with the first partner, an MN fails to complete the pre-handoff and has to switch to a different partner. Handoff delay depends mainly on two broad aspects, the DAD time and the wired and wireless propagation delays needed by the protocol. Since the traditional HMIPv6 does not perform any pre-handoff, the handoff delay is linearly related to the total delay. However, in P_HMIPv6 protocols pre-handoff is essential to be completed early enough to avoid switching to the traditional HMIPv6 delay.

As expected, RSSI method is not performing well even with larger coverage; this is mainly due to the short interconnections with vehicles going to opposite direction or exiting the road segment. LET is not performing well with latencies of 300 ms and 600 ms. Also, we notice using a larger radius worsen LET handoff success rate. VLET is proven to be superior for all latencies.

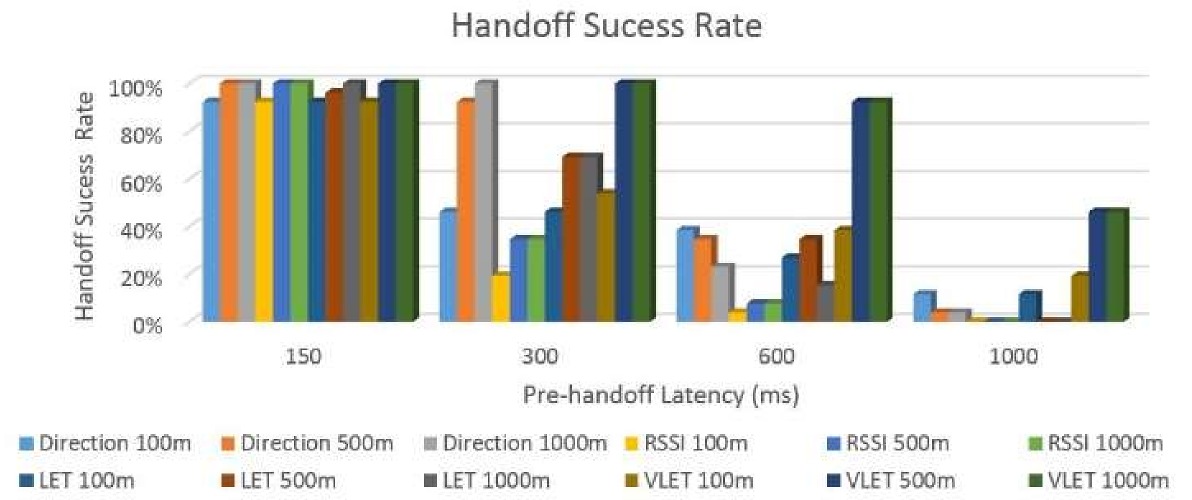


Figure 5.12: Effect of pre-handoff latency on handoff success rate.

5.4.4 Handoff Messaging cost

Liming the number of messages exchanged during the handoff process is crucial to prevent congestion in the areas of handoff. Normally, for each partner selection, there is a message broadcasted to all vehicles in range, and each PN will reply with a message containing the position and speed information. In Fig. 5.13, we show the effect of communication radius on the handoff for Direction, LET, RSSI and VLET. Since the number of PN assisted in handoff is limited, the messages exchanged as well. However, the more range, the more partners are receiving request messages and respond back. Hence, it is normal for the handoff cost to increase with communication radius increase. It is evident that the handoff cost for VLET is much less than all the other protocols, due to the fewer disconnections with partners and the need for requesting new partners.

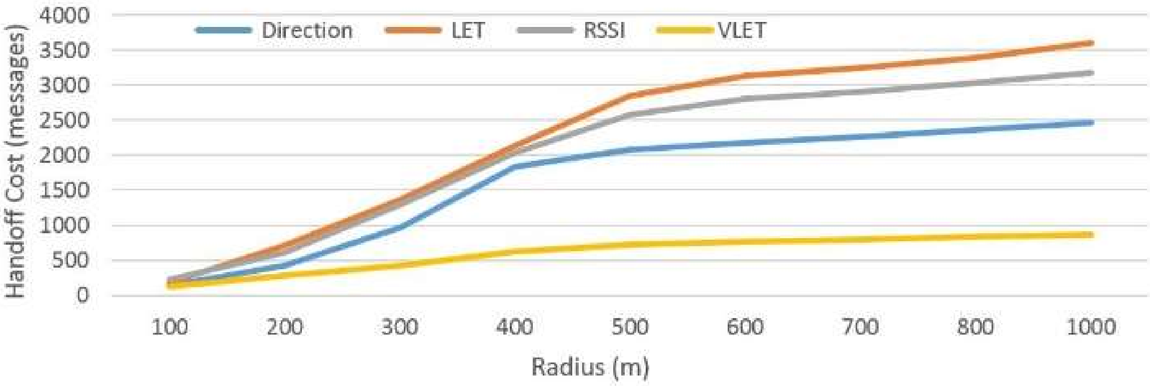


Figure 5.13: Handoff cost with different communication radiuses.

5.5 Conclusion

Partner assisted handoff is proven to enhance the handoff for vehicles, however, most partner selection protocols are not suitable for the vehicular environment. In this paper, we investigated in-depth the partner selection problems on VANET and the reasons behind the failure of the traditional fast MANET partner selections on VANET. We proposed a road topology aware partner selection protocol tailored for VANET. The proposed take advantage of the availability of road topology data to predict the future connectivity period between a roaming and partner vehicles without sharing their future routes. The proposed protocol is compared to the state of the art partner selection protocols. The results show a clear improvement in handoff success rate and handoff cost. The proposed protocol enhanced the average connection time between partner and roaming vehicles by more than 175% as compared to the standard partner selection protocol.

CHAPTER 6 ARTICLE 5: A SELF-ORGANIZED REACTIVE BICASTING HANDOVER FOR LTE NETWORKS

Hazem Ahmed
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
hazem.ahmed@polymtl.ca

Samuel Pierre
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
samuel.pierre@polymtl.ca

Alejandro Quintero
Mobile Computing and Networking Research Laboratory
Department of Computer Engineering
Montréal, Canada
alejandro.quintero@polymtl.ca

Submitted to IEEE Transactions of Vehicular Technologies, 2017.

ABSTRACT

Self-organization and seamless mobility are the keystones of the next generation mobility solutions in cellular networks. The current LTE standards support both macrocells and femtocells which result in more frequent handovers between networks with different QoS. Data forwarding and bicasting are used to avoid service disruption. However, both of them are suffering from QoS

problems. The first suffers from long service disruption time and the later from redundant data. In this paper, we propose a practical self-organized scheme for seamless handover based on optimized reactive bicasting. The innovation of this scheme is the optimization of the time to start and end the bicasting based on the source and target network conditions. Hence, it takes advantage of the bicasting based scheme and minimizing the network overhead. We developed a complete analytical model for evaluating the new scheme and compare it with the standard and state of the art schemes. Numerical results show that using the new scheme significantly decreases the network overhead caused by standard and state of the art bicasting schemes while reducing the service disruption time.

6.1 **Introduction**

The main challenge with using bicasting over forwarding is the extensive use of bandwidth and network resources since the data is sent twice: once to the source and once to the destination for each UE during the handoff. However, bicasting is better than forwarding in the Service Disruption Time (SDT), since the forwarding path from the source to target eNB through the Service Gateway (S-GW) is taking more time than the direct path between the S-GW and the target eNB.

Bicasting is the most efficient way to send data to two different destinations since the data routes are optimized as opposed to sending the data twice. In next generation networks, eNBs are connected using Passive Optical Network (PON)[1], or if there is no direct path between the source and target eNB, the data have to be forwarded using the Service Gateway (S-GW) or a central anchor point. In these cases, bicasting is, evidently, better than forwarding. Jitter is also a problem with the forwarding protocol, where jitter is not an issue for bicasting protocol since all the data are forwarded from S-GW to the destination.

We propose a protocol for enhancing the handoff, called Self-organized proactive bicasting protocol. The main idea of the protocol is to proactively bicast the data to both source and target eNBs. Differently from the state of the art bicasting protocols and the standard LTE bicasting protocol, the bicasting is lasting for a shorter period. Optimizing the bicast period improves the network throughput dramatically and decreases the SDT during the handoff.

The main motives behind this reactive bicasting protocol are to reduce the service interruption time, reduce buffered packets and decrease packet loss. The proposed protocol is activated once a handoff acknowledgment is received by MME. However, the bicasting start will be delayed and the bicasting end will be advanced by periods related to source and target network conditions.

This paper is organized as follows: Section 6.2 introduces the proposed protocol. Section 6.3 explains the related work and state of the art. Section 6.4 introduces the analytical model for modeling SDT, Packet Loss (PL) and Packet Overhead (PO). Section 6.5 investigate the proposed model numerically, and finally, Section 6.6 draws the conclusion

6.2 LTE Handover Protocol Description

In this section, we first present the standard LTE hand-over protocol defined by 3GPP and we show its weak-nesses. Then we introduce our idea behind the pro-posed protocol, then we introduce our proposed proto-col.

6.2.1 Standard LTE Protocol

The LTE standard supports handoff between small cell eNBs by using hard handoff technique where the MME communicates with S-GW to perform path switching. To implement a lossless handoff, the data are either bi-casted or buffered on the target cell until the handoff is complete, as shown in Fig. 6.1.

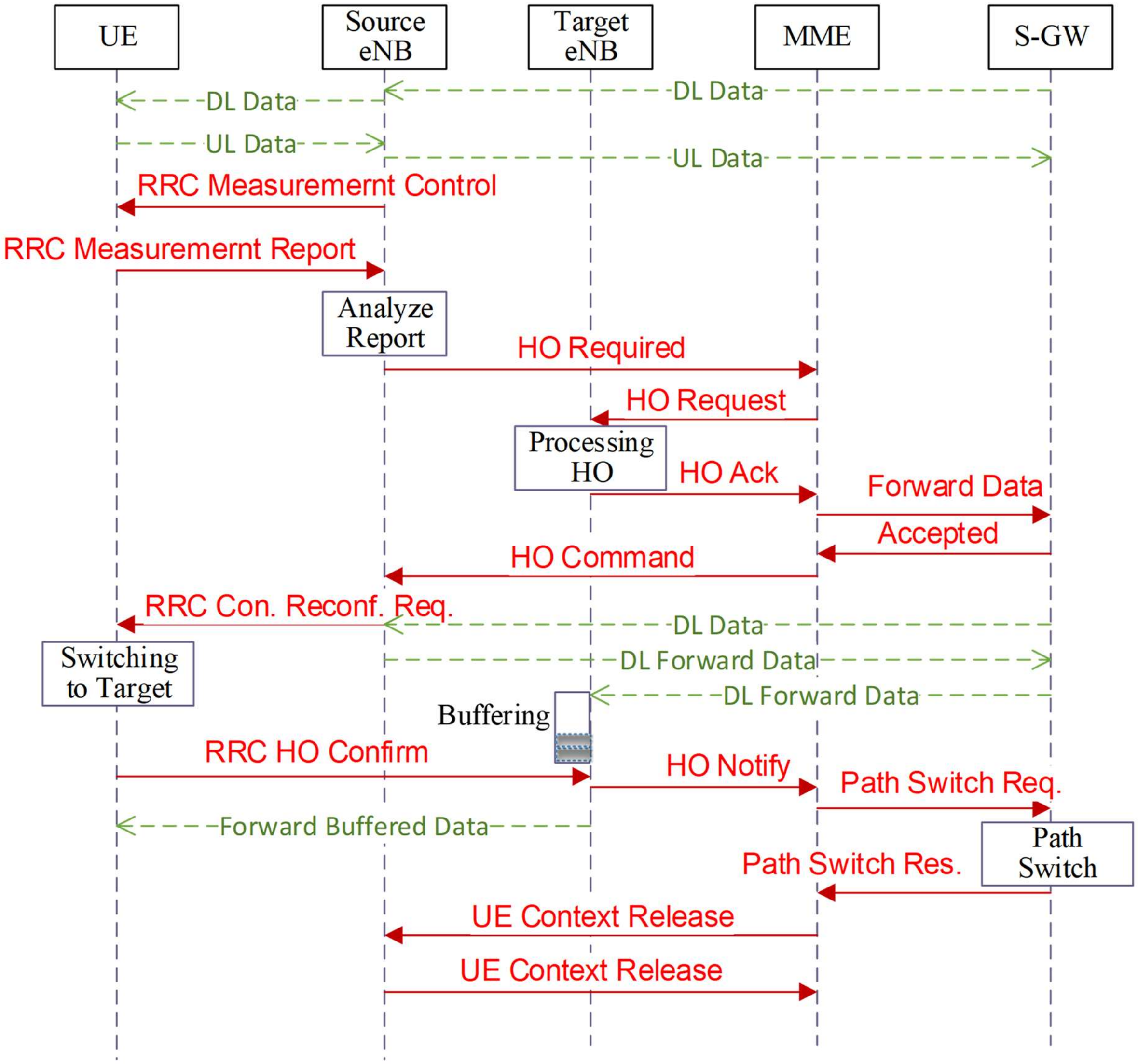


Figure 6.1: Standard LTE Handover

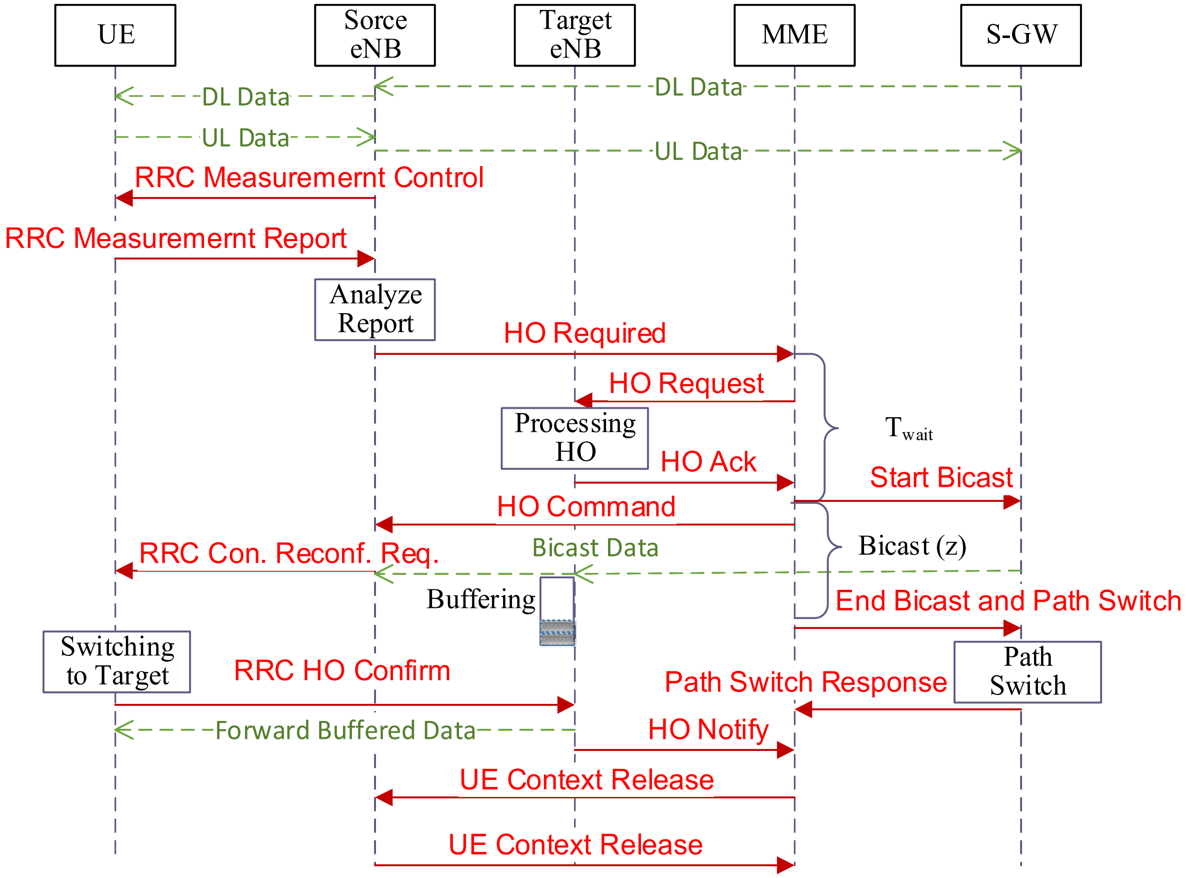


Figure 6.2: Proposed Handover

The UE receives an *RRC Measurement Control* and replays by *RRC Measurement Report* that contains information needed for to select the best Target eNB. Once the report is analyzed, and the handoff is required, the source eNB will send a *Handoff Required* to the MME request to handoff to the Target eNB.

Once the MME receive the *Handoff Required*, it will prepare a *Handover Request* message and send it to the Target eNB. This message contains all required information to create the UE context in the target eNB, including security and bearer information. In or case here the direct forwarding flag will be set to false, then the data will be routed through the S-GW.

As soon as the target eNB received the *Handover Request*, it will check if there are adequate resources to accept the handover session, assign dedicated radio bearer id, allocate radio resources and prepard buffer for downlink data. At this point, the target eNB is ready to buffer any bicasted or forwarded data to the UE. If all the resources are allocated successfully, a *Handover Request Acknowledge* message will be sent to the MME indicating that the assigning resources are completed successfully.

Once the MME receive the *Handover Request Acknowledge* message, it will send a *Handover Command* message to the source eNB to communicate with the UE to start the Handover process. The MME also communicates with the S-GW to establish an indirect data forward tunnel between the source and target eNBs. At this point, all the data are sent to the source eNB is forwarded to the S-GW and then to the target eNB buffer.

The source eNB sends *RRC Connection Reconfiguration* Request to the UE indicating the security algorithm and the target eNB Downlink (DL) and Uplink (UL) ids. Once received by the UE, it will start detaching from the source eNB and Connect to the target eNB. Finally, the UE will send *RRC Connection Reconfiguration Complete* message to the target eNB indicating a successful connection to the target eNB.

The target eNB will send the buffered data, if any, to the UE and will send Handover Notify message to the MME indicating successful handoff and provide the new local home network ID. The MME will communicate with the S-GW to cancel the forwarding tunnel, switch the path the data are directly sent to the target eNB and will send a UE Context Release Command to the source eNB to clean all resources related to UE.

6.2.2 The overall Idea of the proposed protocol

The original LTE forwarding protocol has two drawbacks; the first is that forwarding is consuming so much bandwidth, as seen in Fig. 6.1, the same data are sent three times, i.e. S-GW to Source eNB then Source eNB to S-GW and finally S-GW to target eNB. The second problem is that it is affected by the network latency between eNB and MME since the forwarding cannot start until the negotiation between MME and Target eNB is completed. Moreover, the forwarding will not stop unless messages are communicated from Target eNB to MME.

In order to overcome the forwarding problems, our proposed protocol is using bicasting instead of forwarding. It is worthy to note that, bicasting is mentioned in the standard LTE Handoff. However, it is not recommended due to the significant overhead of bicasting the data during handoff. Our bicasting protocol is different from the standard LTE protocol wherein the later the bicasting is initiated (similar to the forwarding) after MME received *HO Ack* Message. The proposed protocol needs the *Ho Ack* message to neither start nor stop the bicasting. In the proposed protocol once *HO Required* is received, the MME initiates the bicasting after a special timer is expired. This timer is intended to overcome the overhead caused by early bicasting, as shown in Fig. 6.2.

The bicasting lasts for a “z” period, subsequently, the path switch is issued without waiting for *HO Notify* message. Hence the bicasting can be stopped as soon as possible and reduce the packet overhead.

6.2.3 The Proposed Protocol

Similar to the LTE protocol, the UE receives an *RRC Measurement Control* and replies by *RRC Measurement Report* message that contains the information needed to select the best Target eNB. Once the report is analyzed and the handoff is required, the source eNB will send a *Handoff Required* message to the MME request to handoff to the Target eNB.

Once the MME receive the *Handoff Required* message, it will prepare a Handover Request message and send it to the Target eNB. This message contains all required information to create the UE context in the target eNB, including security and bearer information. In this case, the direct forwarding flag will be set to false and the data will be routed through the S-GW. In the meantime, a wait timer is initiated. This timer is used to wait for *T_wait* before starting the data bicasting. The idea behind using this timer is to decrease the amount of data bicasted while the UE still attached to the source eNB. To select and optimize *T_wait* value the delay between MME and eNBs are needed. These values can be achieved by analyzing the *Ho Required* send and receive times at MME to infer the delay.

Once the wait timer expires, the MME sends a *Start Bicast* message to the S-GW to duplicate the data and send them to the target eNB. The bicasting will stop when the MME sends *End Bicast and Path Switch request* to the S-GW. The path switch is performed independently from receiving Handover Notify message from the target eNB. The bicasting period “z” is set to a constant value to cover the expected handover time. For optimizing “z,” a similar function to MRO can be used to report too early or too late bicasting so that the preselected “z” value can be relaxed depending on the network conditions.

As soon as the target eNB receive the *Handover Request*, it checks if there are adequate resources to accept the handover session, assigns dedicated radio bearer identifiers, allocates radio resources and setups buffer for downlink data. At this point, the target eNB is ready to buffer any bicasted or forwarded data to the UE. If all the resources are allocated successfully, the target eNB sends a *Handover Acknowledge* message to the MME indicating that the assigning resources process is completed successfully.

Once the MME receives *the Handover Acknowledge message*, it sends a *Handover Command* message to the source eNB to communicate with the UE to start the Handover process. At this point, all the data is bicasted to the source eNB and to the target eNB where the data are buffered.

The source eNB sends *RRC Connection Reconfiguration Request* message to the UE indicating the security algorithm and the target eNB Downlink (DL) and Uplink (UL) identifiers. Once received by the UE, it starts detaching from the source eNB and connects to the target eNB. Finally, the UE will send *RRC Connection Reconfiguration* Complete message to the target eNB indicating a successful connection to the target eNB.

The target eNB will send the buffered data, if any, to the UE and then sends *Handover Notify* message to the MME indicating successful handoff and provide the new local home network ID. The MME will send a *UE Context Release* Command to the source eNB to clean all resources related to UE.

6.2.4 Handoff Failure Effect

There are many scenarios where a handoff may fail. For example, after sending *HO Request* to the target eNB, it rejects it due to insufficient resources. In the proposed protocol, this is not an issue since by the time Handoff Acknowledge message is received by the MME, the bicasting is not started yet, or it starts for an insignificant period depending on the selection of *T_wait*. Hence there is no packet overhead problem.

In the original LTE protocol, if the handoff is not completed after it is initiated (i.e. *RRC Configuration Confirm* message is never received), the data forwarding will continue until another timer on the MME is expired. All the data forwarded to the false target eNB will be dropped. Moreover, the data forwarded will result in network overhead. However, in the proposed protocol the path is switched much earlier which reduces the network overhead.

6.3 **Related Work**

Reducing the signaling cost and the delay resulting from waiting for path switch in LTE-A networks is investigated in the literature. Guo et al. [11] introduce using a local mobility anchor to work as a proxy for incoming data before and after the handoff. Replacing the relatively far MME by a local anchor reduces the handoff delay. Moreover, it distributes the MME handoff message load on the local anchors.

The results show that using a local mobility anchor decreases the external handoff signaling cost dramatically [11]. However, using a proxy to relay data among neighbors introduces network overhead for not selecting the direct path between S-GW and eNB.

Lin et al. in [12] developed a mathematical model to compare the effect of location update on LTE-A networks. The authors analyzed three different techniques of location updates and their effect on the handover signaling cost.

Self-Organized Network (SON) is an important feature of the future LTE networks, where eNBs can communicate to each other to enhance the network performance, heal from failures (self-healing) and optimize sharing network resources without human intervention [3].

Mobility Robustness Optimization (MRO) is one aspect of SON, where it aims at reducing the number of failed handover for LTE. As an application to MRO, Wegmann et al. [19] provide a protocol to reduce the handover failures in the inter-RAT environment. The Handover KPI parameters are collected from both RATs, to optimize the time to trigger the handoff. Automatic Neighbor Relation (ANR) is another feature of SON where an eNB can request UE to scan neighbor cell information and send it back. Using this feature eNBs can automatically update the neighbor information table and may get the connectivity information from the MME[3].

The effects of forwarding data versus path switching are studied in [104]. Ko et al. [104], introduce an optimized data forwarding protocol that formulates a reward and cost function to optimize the handover process based on the user or network preference (i.e. whether it is delay or throughput sensitive).

Using bicasting for small and macro cells to enhance the handoff is discussed in [13]. The authors introduce a decoupled signaling and user data plans architecture for the high-speed railway for 5G networks. In this architecture, the bicasted user and data plans are not sharing the same frequency bands. However, there is no optimization implemented to reduce the bicast period, and it assumed that the train could connect to both of eNBs at the same time.

To optimize the bicasting period, Dongwook et al. [14] suggest UE using velocity to reduce the bicasting period. By creating different speed profiles, each speed profile is associated with a Signal to Noise Ratio (SNIR) threshold. The results show a reduction in bicasting time. However, their protocol is relying on estimating the UE speed and selecting accurate SNIR value which are not easy tasks and require major modification to the original LTE protocol. Moreover, the bicasting is canceled after the handover is terminated and all communication is completed by MME. Hence, it may last longer based on the network delay between target eNB and MME.

Reactive Bicasting is used to reduce the Service Interruption Time (SIT) in handover between macrocells and femtocells in LTE networks in Guo et al.[15], the authors present a handover protocol where the bicast is initiated by MME once *HO Required* message is received instead of using the traditional forwarding. Since the bicasting starts too early in the handover process, there almost no packet loss and SIT is minimized. One benefit of this protocol is that it does not require any special messages or information from the UE to be shared with the MME.

Using accurate position or estimating it to trigger handover is heavily studied in rail network scenarios, where it is easier to estimate the location due to the fixed railways and it is critical to start the process at high speeds. Usually, rail network protocols are accompanied by using relays to aggregate handover messages from passengers[105]. Huang et al. [106] introduce accurate locations to define a start bicast time for relay-assisted rail networks. The results show a reduction in bicast time for speeds less than 60 m/s as compared to the standard SNIR based and velocity based bicast [14] protocols. However, these methods require UE position estimation which is a challenging issue in networks. Moreover, in the case of macrocells and femtocells, the position has to be accurate to start the bicasting before leaving the boundaries.

All these protocols must wait until the handoff confirmation is received by MME before the bicasting is terminated. The optimal solution is to stop the bicasting exactly when the handoff is completed at the target eNB. However, this is not possible since there is network delay between the target eNB and the MME.

TABLE 6.1, shows a comparison of the proposed algorithm to the state of the art LTE handover protocols. As shown, only the original LTE Bicast protocol does not control the start point of the bicasting. It worth to mention that reactive bicasting is controlling the starting bicast period to be longer to enhance the SIT. Conversely, all the other compared are aiming at reducing bicast period. That is the reason of the high packet overhead.

Table 6.1: Comparison between the proposed protocol and the related work

Protocol	Optimized Start	Optimized End	Network Delay Aware	Packet Overhead	Special Information
Reactive Bicasting [15]	Yes	No	No	Very High	No
LTE Bicast	No	No	No	High	No
Velocity Based [14]	Yes	No	No	medium	Velocity
Position Based [106]	Yes	No	No	medium	Position
LTE Forward	-	-	No	Very High	No
Proposed	Yes	Yes	Yes	Low	No

Both velocity and position based protocols reduce the packet overhead as compared to the standard LTE Bicast technique since they are using special UE information to reduce the starting point of the bicasting.

The proposed protocol is the optimizing both the start and the end of the bicasting period. The end of bicasting period optimization is achieved by proactively performing path switch before the MME receives the *Handover Notify* message. Moreover, using the self-optimization feature of the LTE network, the proposed protocol adapts the waiting timer and the bicast time based on the network delay.

6.4 Analytical Model

In this section, the performance of the proposed protocol, standard LTE, and state of the art protocols are analyzed. Packet overhead is crucial for the performance analysis of the proposed protocol; since packet overhead is the main drawback of using bicasting for LTE[1],[104]. In section 4.1, we analytically analyze the packet overhead of the proposed protocol along with the standard LTE Forwarding and LTE Bicasting ones. Moreover, we included the state of the are reactive bicasting protocol introduced in [15].

As packet loss probability is a critical metric for a handoff protocol, we analytically analyzed the packet loss probability for the proposed protocol in section 6.4.2. Finally, the Service Disruption Time (SDT) is analyzed in section 6.4.3.

6.4.1 Packet Overhead

In the proposed protocol, when the MME receives the *HO Required* message, it starts a wait timer and wait for T_{wait} as show in Fig 6.2. During the T_{wait} data are sent directly to the source eNB, the same way before handoff. In LTE Forwarding, the data is sent to the source eNB only. Once the source eNB receives *HO Command* message, it starts forwarding data to the target eNB. The LTE Bicasting protocol starts bicasting immediately after receiving *HO Acknowledge* message at the MME. However, Reactive Bicasting is starting the bicasting once the *HO Request* message is received by the MME.

Packet overhead has two forms either data bicasting packet overhead or data forwarding packet overhead as shown in Fig. 6.3. For the proposed protocol, the bicasting will last for “z”, and then the path will be switched to the target eNB. However, both LTE Bicasting and Reactive Bicasting protocols will switch to target eNB only after the MME receives *HO Notify* message. Because in the LTE Forwarding the source eNB will keep forwarding incoming data until it receives a context release or no data available at the source to be forwarded.

To model the packet overhead, we use mixed Erlang distribution density function. Erlang distribution is a special case of Gamma distribution that is proven to model real world telecommunication scenarios accurately[107],[108]. The main difference between Gamma and Erlang distributions is that the shape parameter must be an integer as opposed to the Gamma real number generalization.

We assume that the packet arrival rate at the S-GW follows the Poisson arrival process with a rate of $\lambda_{arrival}$ [108]. Also, we assume that *x*, *y*, *h* and *z* are random variables with mixed-Erlang density function similar to [108] and [1] as:

$$f_a(t) = \sum_{r=1}^{N_a} \alpha_{a,r} \frac{\lambda_{a,r}^{m_{a,r}} \, t^{m_{a,r}-1} \, e^{-\lambda_{a,r} t}}{(m_{a,r}-1)!} \quad where \sum_{r=1}^{N_a} \alpha_{a,r} = 1 \quad (1)$$

and its cumulative distribution function (CDF) is

$$F_a(t) = 1 - \sum_{r=1}^{N_a} \alpha_{a,r} \sum_{n=0}^{m_{a,r}-1} \frac{1}{n!} e^{-\lambda_{a,r} t} \left(\lambda_{a,r} t\right)^n \quad (2)$$

The Laplace transformation of (1) is given by:

$$f_a^*(s) = \sum_{r=1}^{N_a} \alpha_{a,r} \left(\frac{\lambda_{a,r}}{s + \lambda_{a,r}}\right)^{m_{a,r}} \quad (3)$$

where a could be x, y, z or h

Let's assume that the S-GW receives start bicasting or forwarding data command at the time τ_0 and it receives the *i*th packet after τ_0 at time τ_i. By using the memoryless property of exponential distribution and from the residual life theorem [109], then $T_i = \tau_i - \tau_0$ Where T_i has an exponential distribution with mean of $\frac{1}{\lambda_{arrival}}$ Thus Ti has an Erlang distribution [108] with density function given by:

$$f_{T_i}(t) = \frac{(\lambda_{arrival}\, t)^{i-1}}{(i-1)!} \lambda_{arrival}\; e^{-\lambda_{arrival} t} \quad (4)$$

And its CDF is given by

$$F_{T_i}(t) = 1 - \sum_{n=0}^{i-1} \frac{1}{n!} e^{-\lambda_{arrival} t} (\lambda_{arrival}\, t)^n \quad (5)$$

And its Laplace transform: $f_{\mathrm{T}_i}^*(s) = \frac{\lambda_{arrival}}{s + \lambda_{arrival}}$ (6)

The expected number of Overhead Packets (N_{OP}) for bicast based protocols is defined as the expected number of packets arrives at the S-GW before the bicast stop command is received.

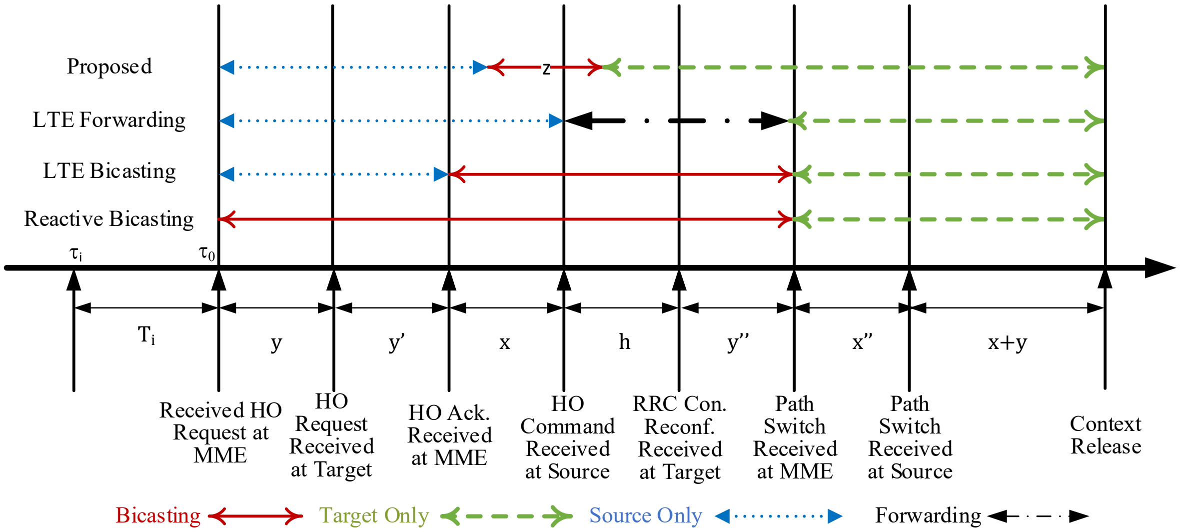


Figure 6.3: Timing diagram comparison between the Reactive Bicasting [1], proposed protocol, LTE Forwarding and LTE Bicasting.

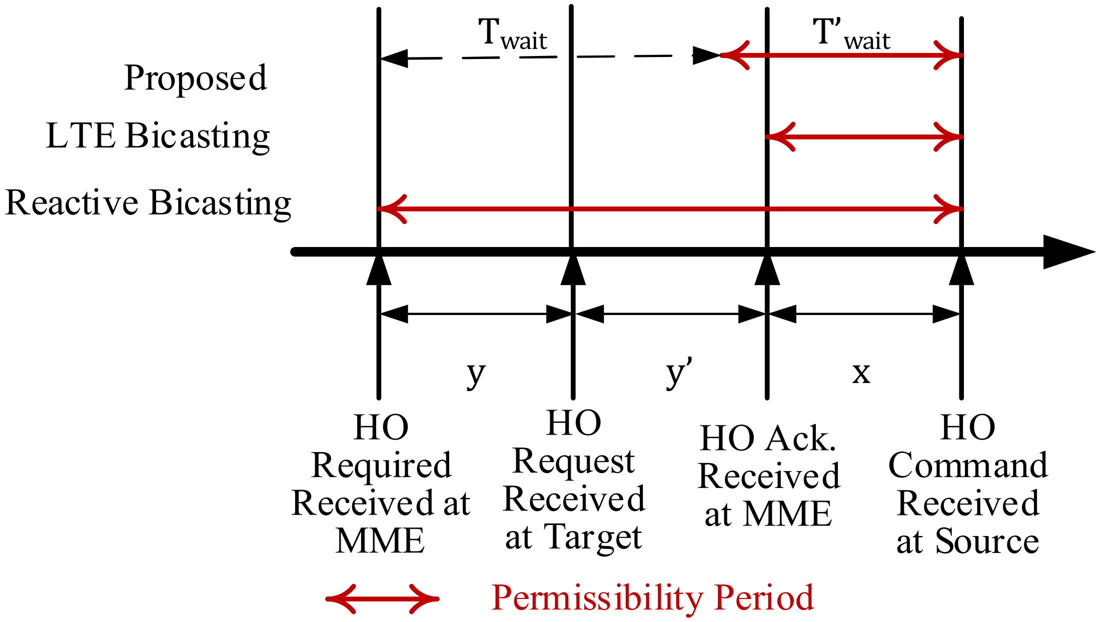


Figure 6.4: The permissibility period for the compared protocols.

However, for the forwarding based, N_{OP} is the period between starting and ending the forwarding at the source eNB.

Let's assume β_i as the ith packet arrives at the S-GW after the bicast start command is received then

$$\beta_i = \begin{cases} 1 & T_i \le \theta \\ 0 & otherwise \end{cases} \quad (7)$$

Where θ is the period between the start and the end of the bicasting.

Then the expected number of overhead packets can be defined as:

$$E[N_{OP}] = \sum_{i=1}^{\infty} \Pr[\beta_i] = \sum_{i=1}^{\infty} \Pr[T_i \le \theta] \quad (8)$$

From Fig. 6.3

$$E^{Re\_bi}[N_{OP}] = \sum_{i=1}^{\infty} \Pr[T_i \le y + y' + x + h + y''] \quad (9)$$

$$E^{LTE\_bi}[N_{OP}] = \sum_{i=1}^{\infty} \Pr[T_i \le x + h + y''] \quad (10)$$

$$E^{Prop}[N_{OP}] = \sum_{i=1}^{\infty} \Pr[T_i \le z] \quad (11)$$

For Erlang $\Pr[T_i \le \theta]$ can be evaluated as

$$\Pr[T_i \le \theta] = \int_{\theta=0}^{\infty} \int_{T=0}^{\theta} f_{T_i}(T)\, f_\theta(\theta) d\theta\, \mathrm{d}T$$

$$= \int_{\theta=0}^{\infty} f_\theta(\theta) \int_{T=0}^{\theta} f_{T_i}(T)\, dT\, d\theta \quad (12)$$

But from the definition of the CDF function

$$\int_{T=0}^{\theta} f_{T_i}(T)\,dT = F_{T_i}(T) \qquad (13)$$

$$\int_{T=0}^{\theta} f_{T_i}(T)\,dT = 1 - \sum_{n=0}^{i-1} \frac{1}{n!}\, e^{-\lambda_{arrival}\,\theta}\, (\lambda_{arrival}\,\theta)^n \quad (14)$$

then

$$Pr[T_i \le \theta] = \int_{\theta=0}^{\infty} f_\theta(\theta) \left[1 - \sum_{n=0}^{i-1} \frac{1}{n!}\, e^{-\lambda_{arrival}\,\theta}\, (\lambda_{arrival}\,\theta)^n\right] d\theta \quad (15)$$

$$= \int_{\theta=0}^{\infty} f_\theta(\theta)\; d\theta - \int_{\theta=0}^{\infty} f_\theta(\theta) \sum_{n=0}^{i-1} \frac{1}{n!}\, e^{-\lambda_{arrival}\,\theta}\, (\lambda_{arrival}\,\theta)^n\; d\theta \quad (16)$$

$$= 1 - \sum_{n=0}^{i-1} \frac{\lambda_{arrival}^n}{n!} \int_{\theta=0}^{\infty} f_\theta(\theta)\; e^{-\lambda_{arrival}\,\theta}\, \theta^n\, d\theta \quad (17)$$

$$= 1 - \sum_{n=0}^{i-1} \frac{\lambda_{arrival}^n}{n!} \left[(-1)^n \frac{d^n}{ds^n}\, f_\theta^*(s)\right]_{s=\lambda_{arrival}} \quad (18)$$

$$E[N_{OP}] = f_\theta^*(s)\big|_{s=0} - \lambda_{arrival} \frac{d}{ds}\, f_\theta^*(s)\Big|_{s=0} \quad (19)$$

For proof please refer to Appendix B.

From (9), (10) and (11)

$$f_\theta^{Re\_bi*}(s) = f_x^*(s)\, \left(f_y^*(s)\right)^3 f_h^*(s) \quad (20)$$

$$f_\theta^{LTE\_bi*}(s) = f_x^*(s)\, f_h^*(s) f_y^*(s) \quad (21)$$

$$f_\theta^{proposed*}(s) = f_z^*(s) \quad (22)$$

However for LTE forwarding protocol, the data will be forwarded if only it reaches the source. For example, if the data is sent to the source eNB before the *HO Command* or after the path switch at S-GW the data will not be forwarded and hence will be no overhead, then

$$E^{LTE\_Fr}[N_{OP}] = \sum_{i=1}^{\infty} \Pr\left[\, T_i + x \ \leq h + y''\right] \qquad (23)$$

Assume $Q_i = T_i + x$ and $\mu = h + y''$

$$\Pr(\, Q_i \leq \ \mu) \ = \int_{Q_i=0}^{\infty} f_{Q_i}(Q) \int_{t=Q}^{\infty} f_\mu(t)\, dt\, \mathrm{d}Q \quad (24)$$

$f_\mu(t)$ is defined as a summation of two random variables with mixed Erlang distribution. For derivation of $f_\mu(t)$, please refer to Appendix A.

Firstly, for identical rate case using (A.5)

$$\int_{t=Q}^{\infty} f_\mu(t)\, dt \ = \sum_{o=1}^{N_1} \alpha_{h,o}\, \alpha_{y,o}\, \frac{(\lambda_{h,o}\ t)^{m_{h,o}+m_{y,o}-1}}{(m_{h,o}+m_{y,o}-1)!}\ \lambda_{h,o}\, e^{-\lambda_{h,o}\, t}$$

$$= \sum_{o=1}^{N_h} \alpha_{h,o}\, \alpha_{y,o} \sum_{k=0}^{(m_{h,o}+m_{y,o}-1)} \frac{1}{k!}\, e^{-\lambda_{h,o}\, Q} \left(\lambda_{h,o} Q\right)^k \quad (25)$$

$$\Pr(\, Q_i \leq \ \mu)$$

$$= \sum_{o=1}^{N_h} \alpha_{h,o}\, \alpha_{y,o} \sum_{k=0}^{(m_{h,o}+m_{y,o}-1)} \frac{1}{k!} \left(\lambda_{h,o}\right)^k \int_{Q_i=0}^{\infty} f_{Q_i}(Q)\, Q^k\, e^{-\lambda_{h,o}\, Q}\, dQ \quad (30)$$

$$\Pr(\, Q_i \leq \ \mu) = \sum_{o=1}^{N_h} \alpha_{h,o}\, \alpha_{y,o} \sum_{k=0}^{(m_{h,o}+m_{y,o}-1)} \frac{1}{k!} \left(\lambda_{h,o}\right)^k \left[(-1)^K \frac{d^K}{ds^K}\, f_{Q_i}^*(S)\right]_{s=\lambda_{h,o}}$$

$$for\ \lambda_{h,o} = \lambda_{y,j} \qquad (31)$$

Secondly, for different rate case for $f_\mu(t)$, using (A.7)

$$\int_{t=Q}^{\infty} f_{\mu}(t)\,dt \;=\; \sum_{o=1}^{N_h}\sum_{j=1}^{N_y} \alpha_{h,o}\,\alpha_{y,j}\left[\int_{t=Q}^{\infty} g_{h,o}(t)\,dt + \int_{t=Q}^{\infty} g_{y,j}(t)\,dt\;\right] \qquad (32)$$

$$\int_{t=Q}^{\infty} g_{h,o}(t)\,dt = \;\lambda_{h,i}^{m_{h,o}} \sum_{k=1}^{m_{h,o}} \varphi_{h,k,o}\,\frac{(-1)^{m_{h,o}-k}}{(k-1)!}\;\int_{t=Q}^{\infty} e^{-\lambda_{h,o}t}t^{k-1}\,dt \qquad (33)$$

But using the upper incomplete Gamma function definition

$$\int_{t=Q}^{\infty} e^{-\lambda_{h,o}t}t^{k-1}\,dt \;=\frac{(k-1)!}{\lambda_{h,o}^{k}}\,e^{-\lambda_{h,o}Q}\sum_{w=0}^{k-1}\frac{Q^{w}\,\lambda_{h,o}^{w}}{w!} \qquad (34)$$

$$G_{h,i}(Q) = \int_{t=Q}^{\infty} g_{h,o}(t)\,dt$$

$$=\;\lambda_{h,o}^{m_{h,o}} \sum_{k=1}^{m_{h,o}} \frac{(-1)^{m_{h,o}-k}}{\lambda_{h,o}^{k}}\;\sum_{w=0}^{k-1}\frac{Q^{w}\,\lambda_{h,o}^{w}}{w!}\;e^{-\lambda_{h,o}Q}\varphi_{h,k,o} \qquad (35)$$

$$\Pr(\,Q_i \le\;\mu) = \int_{Q_i=0}^{\infty} f_{Q_i}(Q)G_{h,o}(Q)\,dQ + \int_{Q_i=0}^{\infty} f_{Q_i}(Q)G_{y,j}(Q)\,dQ \qquad (36)$$

$$\int_{Q_i=0}^{\infty} f_{Q_i}(Q)G_{h,o}(Q)\,dQ$$

$$= \lambda_{h,o}^{m_{h,o}} \sum_{k=1}^{m_{h,o}} \frac{(-1)^{m_{h,o}-k}}{\lambda_{h,o}^{k}}\;\varphi_{h,k,o}\sum_{w=0}^{k-1}\frac{\lambda_{h,o}^{w}}{w!}\;\int_{Q_i=0}^{\infty} f_{Q_i}(Q)\,Q^{w}e^{-\lambda_{h,o}Q}\,dQ \qquad (35)$$

$$= \lambda_{h,o}^{m_{h,o}} \sum_{k=1}^{m_{h,o}} \frac{(-1)^{m_{h,o}-k}}{\lambda_{h,o}^{k}}\,\varphi_{h,k,o}\,\sum_{w=0}^{k-1}\frac{\lambda_{h,o}^{w}}{w!}\;\left[(-1)^{w}\frac{d^{w}}{ds^{w}}\,f_{Q_i}(s)\right]_{s=\lambda_{h,o}} \qquad (36)$$

$$\Pr(\, Q_i \leq \, \mu)$$

$$= \sum_{o=1}^{N_h} \sum_{j=1}^{N_y} \alpha_{h,o}\, \alpha_{y,j} \left[\lambda_{h,o}^{m_{h,o}} \sum_{k=1}^{m_{h,o}} \frac{(-1)^{m_{h,o}-k}}{\lambda_{h,o}^{k}} \, \varphi_{h,k,o} \sum_{w=0}^{k-1} \frac{\lambda_{h,o}^{w}}{w!} \, \left[(-1)^{w} \frac{d^{w}}{ds^{w}} \, f_{Q_i}^{*}(S) \right]_{s=\lambda_{h,o}} \right.$$

$$\left. + \, \lambda_{y,j}^{m_{y,j}} \sum_{k=1}^{m_{h,j}} \frac{(-1)^{m_{y,j}-k}}{\lambda_{y,j}^{k}} \, \varphi_{y,k,j} \sum_{w=0}^{k-1} \frac{\lambda_{y,j}^{w}}{w!} \, \left[(-1)^{w} \frac{d^{w}}{ds^{w}} \, f_{Q_i}^{*}(S) \right]_{s=\lambda_{y,j}} \right] \quad (37)$$

$$and \; f_{Q_i}(s) = \; f_{T_i}^{*}(s) \, f_{x}^{*}(s)$$

6.4.2 Packet Loss

Packet loss is a very important aspect of handoff. In case of LTE forwarding protocol, there is almost no packet loss due to the fact the data packets are always sent from S-GW to source eNB and then forwarded to the target eNB. However for bicasting protocols, packets will be lost if they arrive the S-GW before the *start bicasting* command and they reache the source eNB after the UE is already disconnected from the source eNB.

The timing diagram for lost packets is shown in Fig. 6.4. We define Permissibility Period (PP) as the time from receiving the bicasting command until the source eNB receives the *Ho Command*, and the UE is detached. For the proposed protocol, packets will be lost if it arrives at the S-GW within or before T_{wait} only. After T_{wait}, the data will be bicasted, received and buffered by target eNB. If T_{wait} equals to zero, then PP will be the same as Reactive Bicasting. However, if T_{wait} equals exactly y + y’ then the proposed packet loss will be the same as LTE Bicasting. Similar to the buffered packet analysis we will use T'_{wait} to represent the reminder of T_{wait}.

Similar to [15] and [108] we will use P_i as the period between the time of *i*th packet received at S-GW before the bicast command is received. P_i has an Erlang distribution with density function defined in (1).

From Fig. 6.4, the maximum packet loss can be calculated as the following:

$$R_i^{Re\_bi} = \; y + y' + x + P_i \qquad (38)$$

$$PL_{Max}^{Re\_bi} = \sum_{i=1}^{\infty} \Pr\left[R_i^{Re\_bi} \leq x_i\right] \qquad (39)$$

$$Pr\left[R_i^{Re\_bi} \leq x_i\right] = \int_{R=0}^{\infty} \int_{x=R}^{\infty} f_{R_i^{Re\_bi}}(R)\, f_{x_i}(x) dx\, \mathrm{d}R \quad (40)$$

$$= \sum_{r=1}^{N_x} \alpha_{x,r} \sum_{n=0}^{m_{x,r}-1} \frac{\lambda_{x,r}^n}{n!} \left[(-1)^n \frac{d^n}{ds^n} f_{R_i^{Re_{bi}}}^*(s)\right]_{s=\lambda_{x,r}} \quad (41)$$

where

$$f_{R_i^{Re\_bi}}^*(S) = \left(f_y^*(S)\right)^2 . f_x^*(S) . f_{P_i}^*(S) \qquad (42)$$

$$PL_{Max}^{Re\_bi} = \sum_{i=1}^{\infty} \sum_{r=1}^{N_x} \alpha_{x,r} \sum_{n=0}^{m_{x,r}-1} \frac{\lambda_{x,r}^n}{n!} \left[(-1)^n \frac{d^n}{ds^n} f_{R_i^{Re_{bi}}}^*(s)\right]_{s=\lambda_{x,r}} \quad (43)$$

$$R_i^{LTE\_bi} = x + P_i \quad (44)$$

$$PL_{Max}^{LTE\_bi} = \sum_{i=1}^{\infty} \Pr\left[R_i^{LTE\_bi} \leq x_i\right] \qquad (45)$$

Similar to (43) maximum packet loss for LTE_BI is

$$PL_{Max}^{LTE\_bi} = \sum_{i=1}^{\infty} \sum_{r=1}^{N_x} \alpha_{x,r} \sum_{n=0}^{m_{x,r}-1} \frac{\lambda_{x,r}^n}{n!} \left[(-1)^n \frac{d^n}{ds^n} f_{R_i^{LTE\_bi}}^*(s)\right]_{s=\lambda_{x,r}} \quad (46)$$

$$where\ f_{R_i^{LTE\_bi}}^*(s) = f_x^*(s) . f_{P_i}^*(s) \quad (47)$$

For the proposed protocol, assume

$$R_i^{proposed} = T'_{wait} + P_i \qquad (48)$$

$$PL_{Max}^{Proposed} = \sum_{i=1}^{\infty} \Pr\left[R_i^{proposed} \leq x_i\right] \qquad (49)$$

$$f^*_{R_i^{proposed}}(s) = f^*_{T'_{wait}}(s) \, . \, f^*_{P_i}(s) \qquad (50)$$

6.4.3 Service Disruptions Time

The Service Disruption Time (T_{sdt}) is defined as the period between detaching from the source eNB and receiving data from the target eNB. In bicasting protocols, if there are data already buffered at target eNB the UE starts receiving data immediately, in this case T_{sdt} is equal to h. The general rule for T_{sdt} for bicasting is defined as the maximum between h and l , where l is a random variable represents the period of time taken to receive bicasting data at target eNB before the UE detaches from the source eNB, then the expected service distribution time is

$$E[T_{sdt}] = \max(E[l], E[h]) \begin{cases} E[h] & E[h] \geq E[l] \\ E[l] & otherwise \end{cases} \qquad (51)$$

And

$$E[h] = \left(-\acute{f}^*_h \, (s)\Big|_{s=0} \right) \qquad (52)$$

For the proposed protocol l can be calculated as the difference between the time required to detach from source eNB and T_{wait} and assuming the latency between MME and target eNB to be y'' then

$$l^{proposed} = y'' - \left((y + y' + x) - T_{wait} \right) \quad (53)$$

$$l^{proposed} = T_{wait} + y'' - (y + y' + x) \qquad (54)$$

$$E[l^{proposed}] = E[T_{wait}] + E[\mathrm{y}''] - E[\mathrm{y}] - E[\mathrm{y}'] - E[\mathrm{x}] \quad (55)$$

Since y, y' and y'' are i.i.d. then

$$E[l^{proposed}] = E[T_{wait}] - E[\mathrm{y}] - E[\mathrm{x}] \qquad (56)$$

Similarly for the reactive bicasting, the $E[T_{wait}]$= 0

$$l^{Re\_bi} = y'' - (\mathrm{y} + \mathrm{y}' + \mathrm{x}) \qquad (57)$$

$$E\left[l^{Re\_bi}\right] = -E[\mathrm{y}] - E[\mathrm{x}] \qquad (58)$$

For LTE bicasting protocol as shown in Fig. 6.3, the time available before UE detach is x, then

$$l^{LTE\_bi} = y'' - x \qquad (59)$$

$$E\left[l^{LTE\_bi}\right] = E[\mathrm{y}] - E[x] \qquad (60)$$

For LTE forwarding[15], the data must be forwarded from source eNB to target eNB then

$$l^{LTE\_Fr} = y + x \qquad (61)$$

$$E[l^{LTE\_Fr}] = E[\mathrm{y}] + E[x] \qquad (62)$$

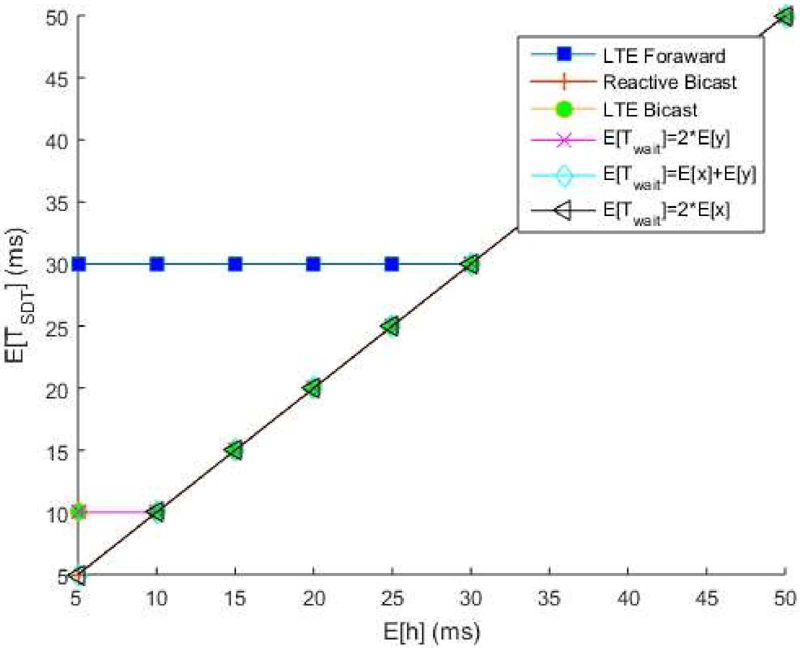
6.5 Performance Evaluation

In this section, we evaluate the numerical results for the developed mathematical model. Similar to [15] and [108] we will use $\alpha_{a,i} = 0.5$, $N_a = 2$ and $m_{a,i} = 2\ for\ i = 1,2$ where a represents $x, y, z, h\ or\ T_{wait}$. MATLAB is used to produce the numerical results from the equations in the analytical model. In our numerical analysis, we consider both lower source latency case and lower destination latency case. Lower source latency is usually the case for smaller cell to bigger one (e.g. femto cell to macro cell) and the opposite for lower destination latency. Also, we considered the effect of the device handoff latency and the relative network delay effects on each of SDT, packet loss and packet overhead. For lower source latency we will use $\lambda_{1,x} = 300\ and\ \lambda_{1,y} = 150$ and for lower destination latency we will use $\lambda_{1,x} = 150\ and\ \lambda_{1,y} = 300$. If otherwise mentioned, we use $\lambda_{1,h} = 300$. In our evaluation we used three different settings for setting for $E[T_{wait}]$, the first depends only on the destination latency where $E[T_{wait}] = 2 * E[y]$, the second depends only on the source latency $E[T_{wait}] = 2 * E[x]$, and the third depends on both source and destination latencies where $E[T_{wait}] = E[x] + E[y]$.

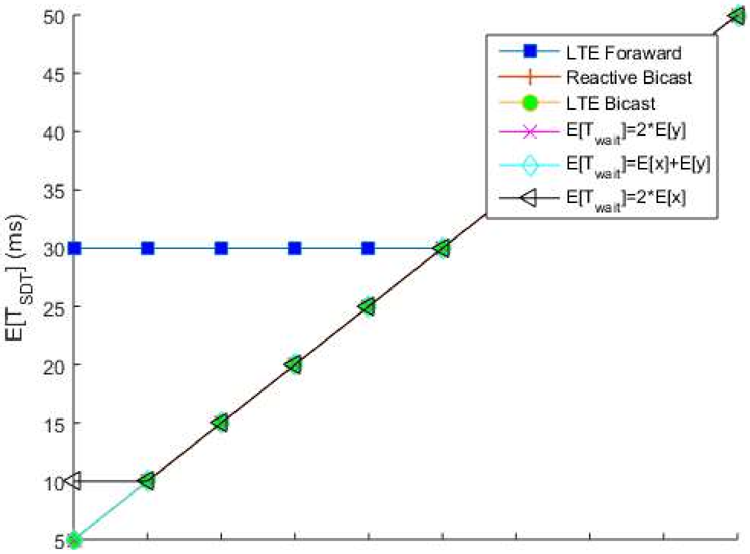
6.5.1 Service Disruptions Time

Fig. 6.5, illustrates the effect of the UE handoff latency $E[h]$ on the expection of Service Disruption Time $E[T_{sdt}]$ as analyzed in equation (51). For both lower source and destination latencies, $E[T_{sdt}]$ depends mainly on $E[\mathrm{h}]$ for all the protocols after $E[h] > 30\ ms$ which is the summation of both source and destination expected latencies. Moreover, the forwarding protocol gives the worse performance since it depends on both source and destination latencies. In Fig. 6.5 (a) it is shown that both LTE Bicast and setting $E[T_{wait}] = 2 * E[\mathrm{y}]$ produces higher SDT when $E[\mathrm{h}] < E[x]$.

Similarly in Fig. 6.5 (b) setting $E[T_{wait}] = 2 * E[\mathrm{x}]$ produces higher SDT when $E[\mathrm{h}] < E[x]$. The rest are performing very well for all values of $E[\mathrm{h}]$.



(a) Source latency is less than destination latency

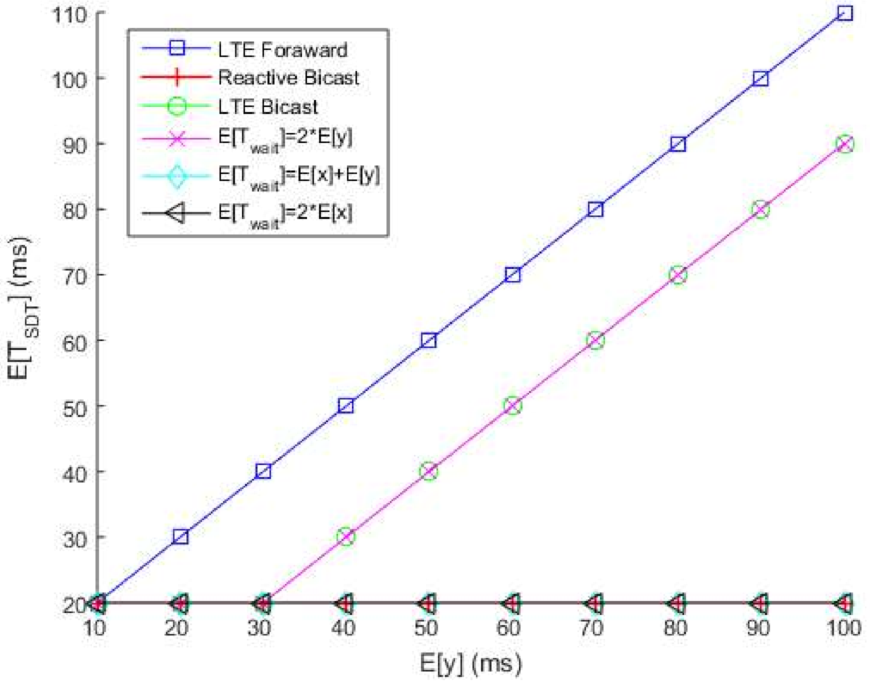


(b) Destination latency is less than source latency

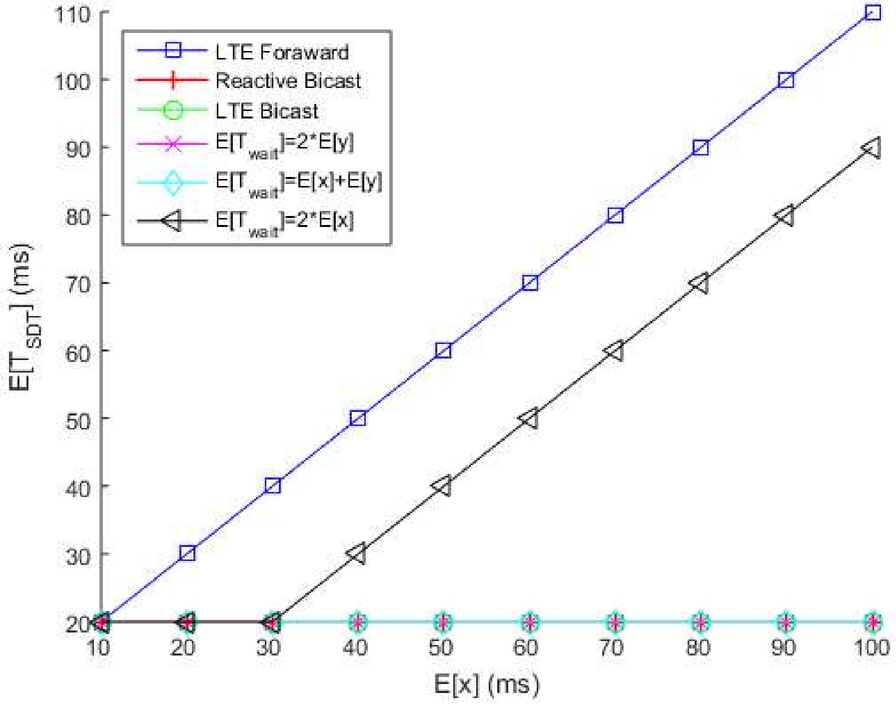
Figure 6.5: Effect of device handoff latency on SDT

Fig. 6.6, shows the effect of (a)destination latency increase and (b) source latency increase. It is clear that the LTE Forwarding protocol behaves badly for both (a) and (b). The minimum in both graphs is 20 ms which is due to the $E[\mathrm{h}]$. In Fig. 6.6 (a), both LTE Bicast and setting $E[T_{wait}] = 2 * E[\mathrm{y}]$ are stable for lower latency but will increase as $E[\mathrm{y}]$ increases. Similarly in Fig. 6.6 (b), when setting $E[T_{wait}] = 2 * E[\mathrm{x}]$.

Fig. 6.7, illustrates the effect of overall network delay increase. As shown, LTE Forward protocol is much worse than all the other protocols. All the other protocols will keep stable at the minimum even with network delay increase of 200%. This mainly due to the depency of the LTE fowarding on both source and destination latencies.

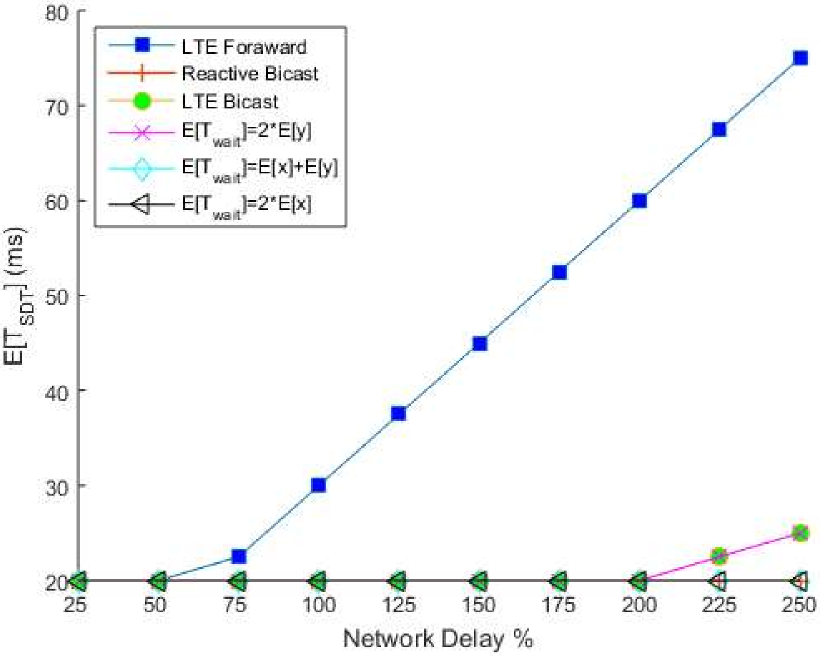


(a) Effect of destination latency on SDT

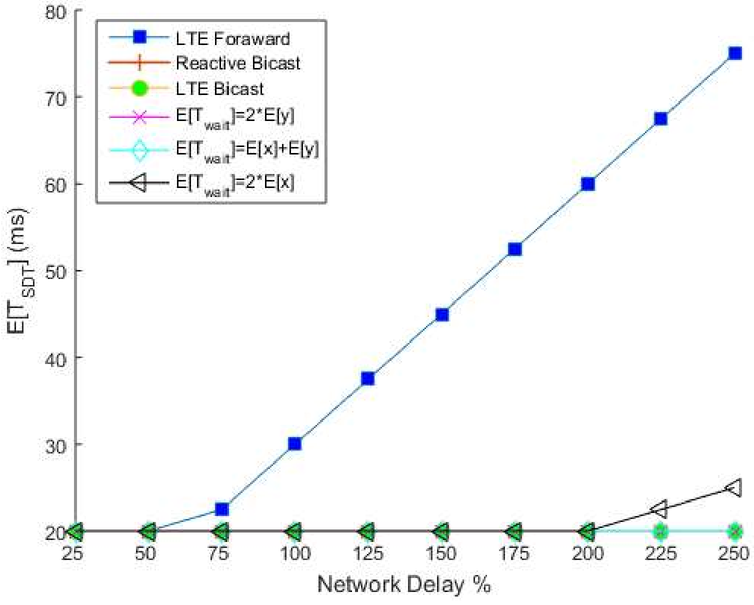


(b) Effect of source latency on SDT

Figure 6.6: Effect of latencies on the SDT



(a) Source latency is less than destination latency

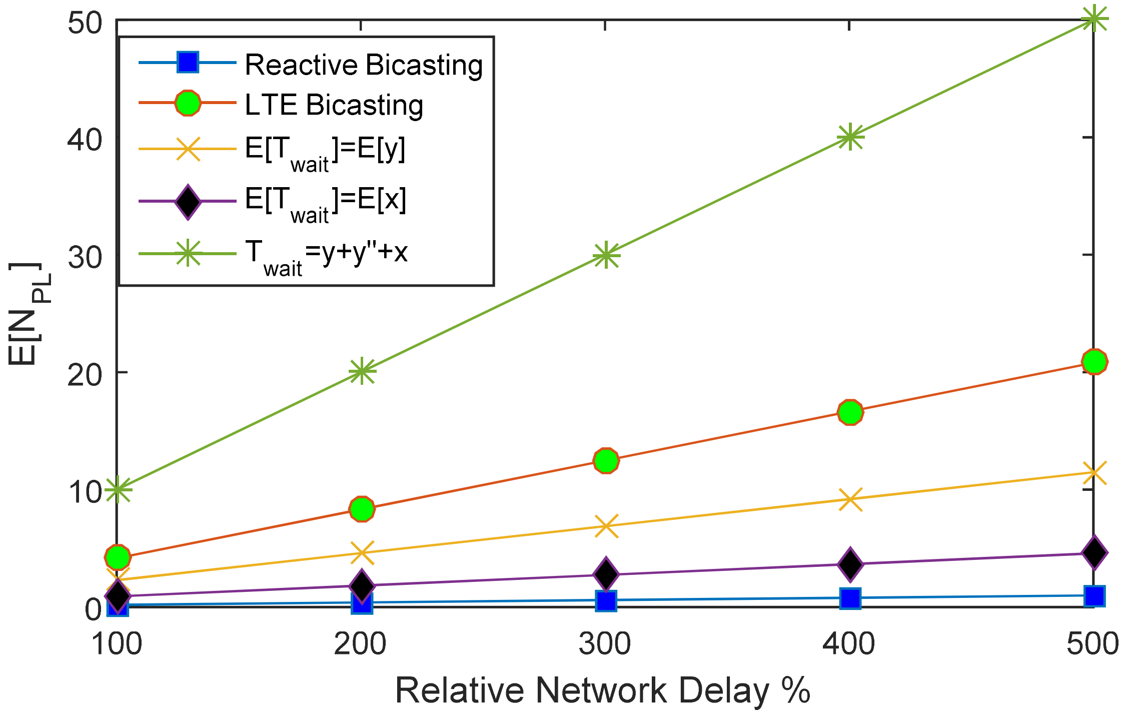


. (b) Destination latency is less than source latency

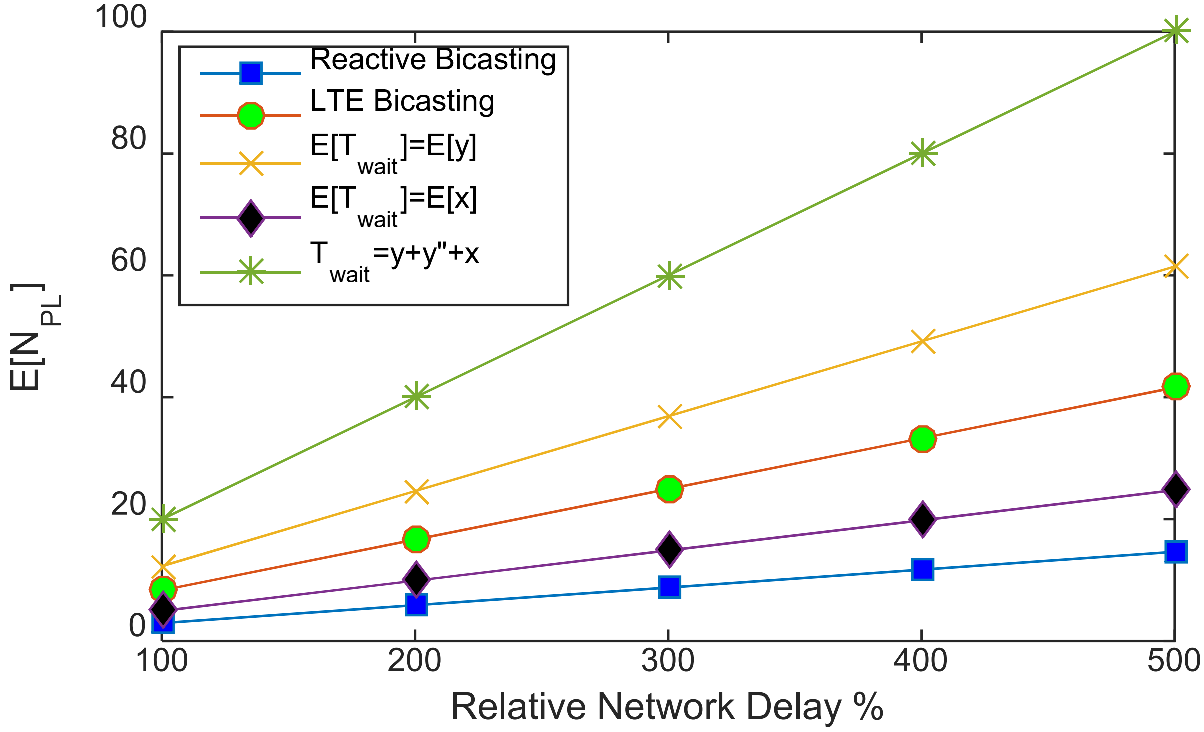
Figure 6.7: Effect of network delay increase on the SDT

6.5.2 Packet Loss

The expected value of the number of packet loss due to handoff $E[N_{OP}]$ is relative to the overall network delay. The overall network delay may increase in specific periods of the day due to heavy traffic. As shown in Fig. 6.8, the earlier the bicast starts, the lower the packet loss. The Reactive Bicasting protocol performs the best due to the very early start of bicasting. In the proposed protocol, setting T_{Wait} to the source network delay $E[x]$ performs very well for both lower destination and lower source latencies. The reason is that for the data to be lost it has to be received at the source first. However, setting the T_{Wait} to be the maximum of y+y"+x is increasing the probability for packet loss.



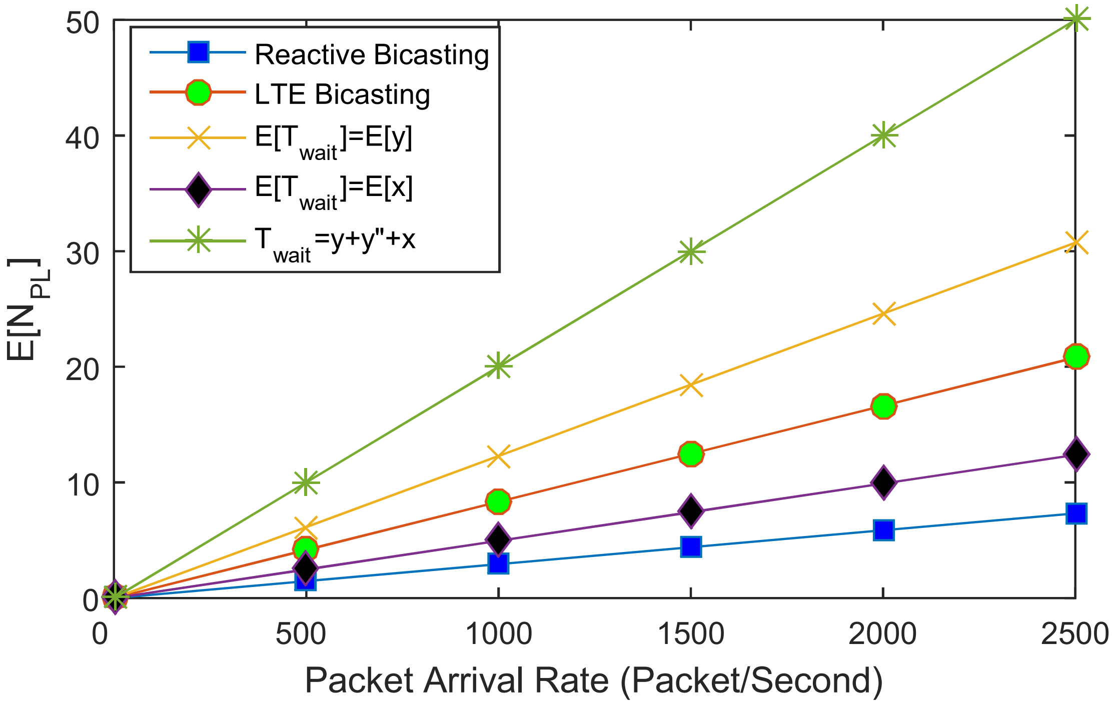
(a) Source latency is less than destination latency



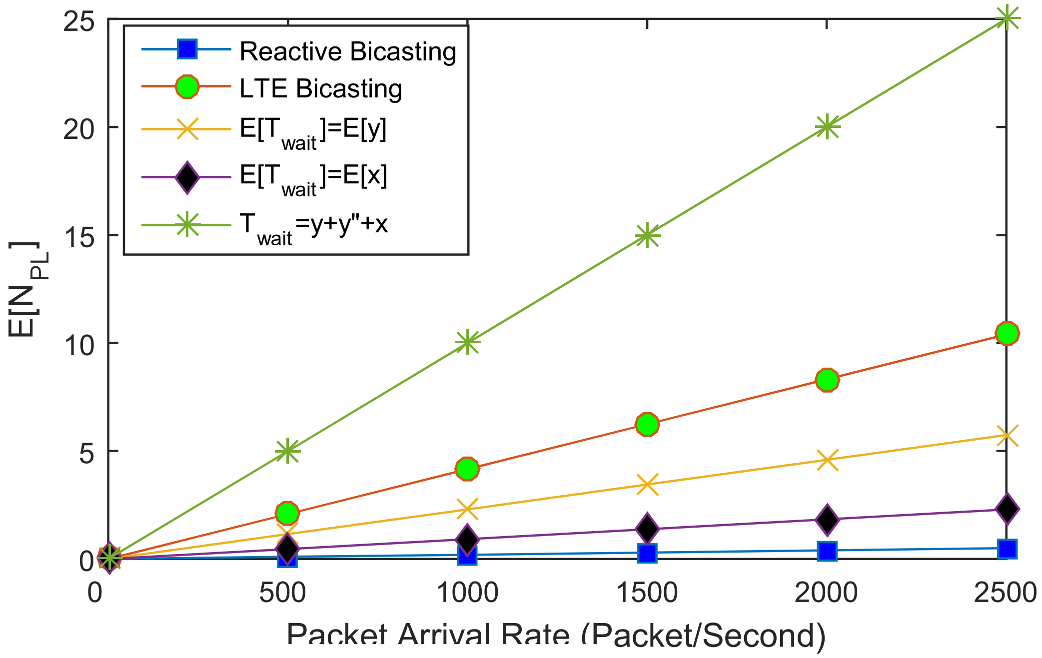
(b) Destination latency is less than source latency

Figure 6.8: Effect of network delay increase on the expected packet loss

In Fig. 6.9, the effect of increasing packet arrival rate on packet loss is shown. The more the packet arrival rate the more the packet loss. Hence, for using $E[N_{OP}] = E[\text{x}]$ provides a good results for



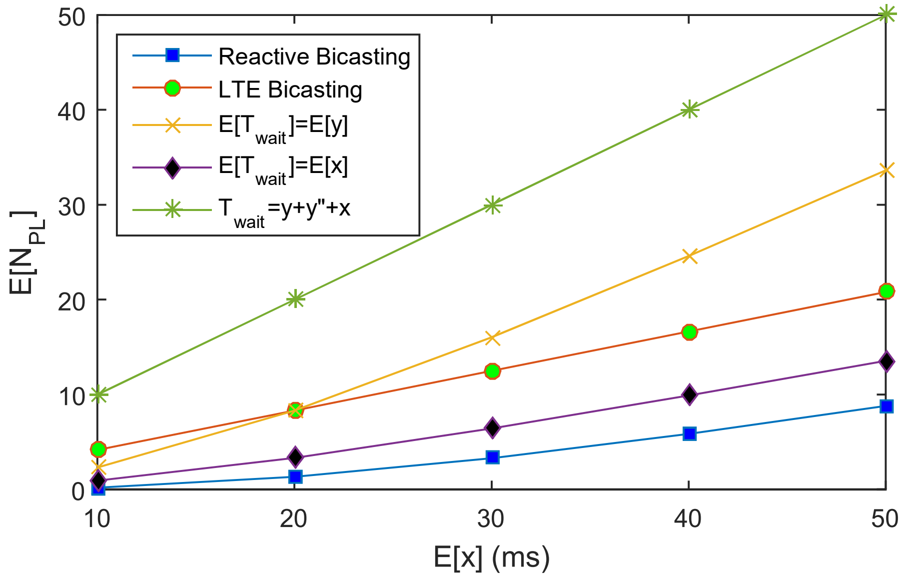
(a) Source latency is less than destination



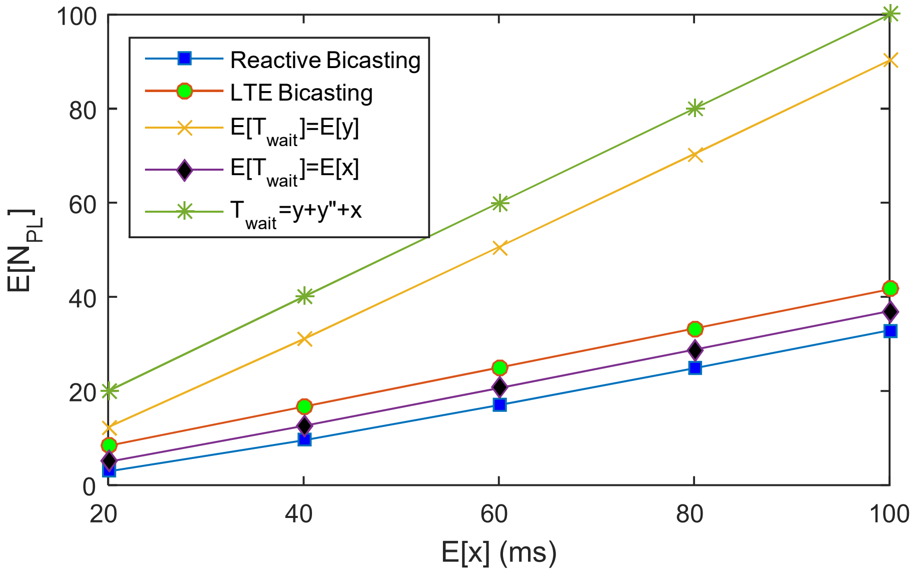
(b) Destination latency is less than source

Figure 6.9: Effect of packet arrival rate on packet loss

In Fig. 6.10 (a), we show the effect of increasing source latency. As shown, for both (a) and (b) the more the source latency, the more the packet loss. However, when using T_{Wait} is depending on the destination latency the packet loss probability becomes very sensitive to the change in source latency (i.e. it increases much faster with any source latency increase).



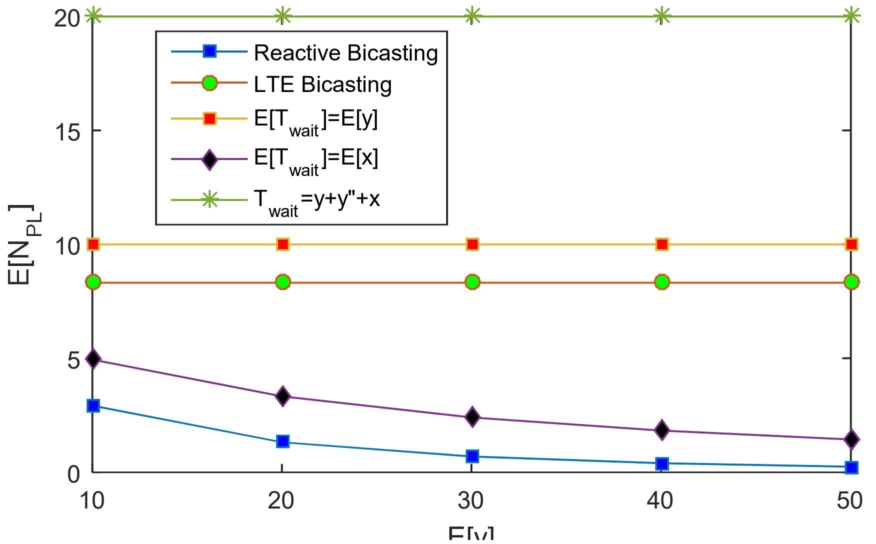
(a) Source latency is less than destination



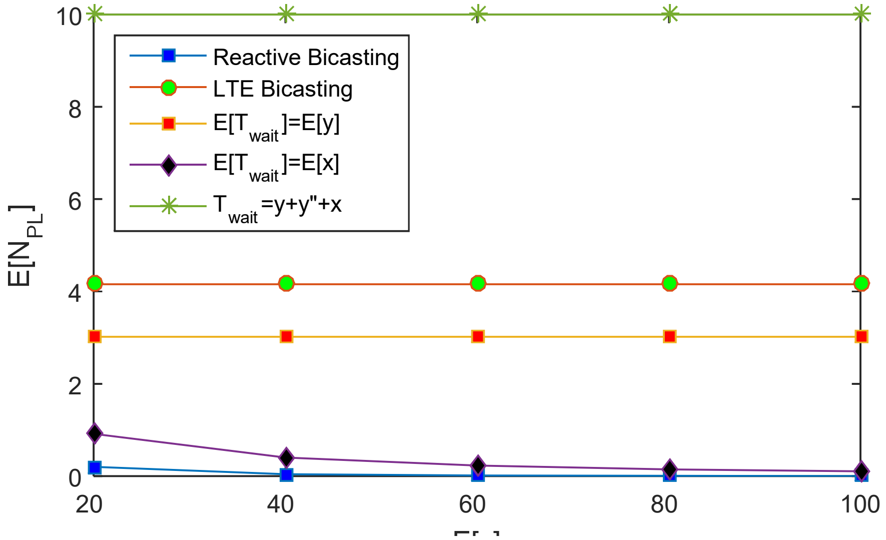
(b) Destination latency is less than source

Figure 6.10: Effect of source latency on the expected packet loss

In Fig. 6.11, we show the effect of the change in destination delay. LTE Bicasting and using $E[T_{Wait}]$ equals to E[y] both do not change much. However, Reactive Bicasting and using $E[T_{Wait}]$ equals to E[X] are decreasing as the E[y] increases. Since when the E[y] increases, the time that takes packets to reach the destination is more, hence lower expected packet loss.



(a) Source latency is less than destination latency



(b) Destination latency is less than source

Figure 6.11: Effect of destination latency on the expected packet loss

6.5.3 Packet Overhead

Packet overhead depends mainly on the period of time the bicasting or forwarding will be performed and the packet arrival rate. In Fig. 6.12, we show the effect of packet arrival rate on the expected value of the number of overhead packets E[N_{OP}]. Reactive Bicasting overhead increases dramatically with the increase of packet overhead because of the long period of bicasting. However, the LTE Forwarding is the lowest due to the short time of data forwarding. Using LTE Bicasting results in high overhead, due to the continuity of bicasting waiting for the path switch. It is obvious that all the proposed protocols performs well, since both the start of the bicasting and the end of it are optimized.

In Fig. 6.13, we show the effect of device handoff latency on the E[N_{OP}]. Again, Reactive Bicasting and LTE Bicasting are the highest in overhead. The LTE Forwarding depends on the time taken for handoff as opposed to the proposed protocols.

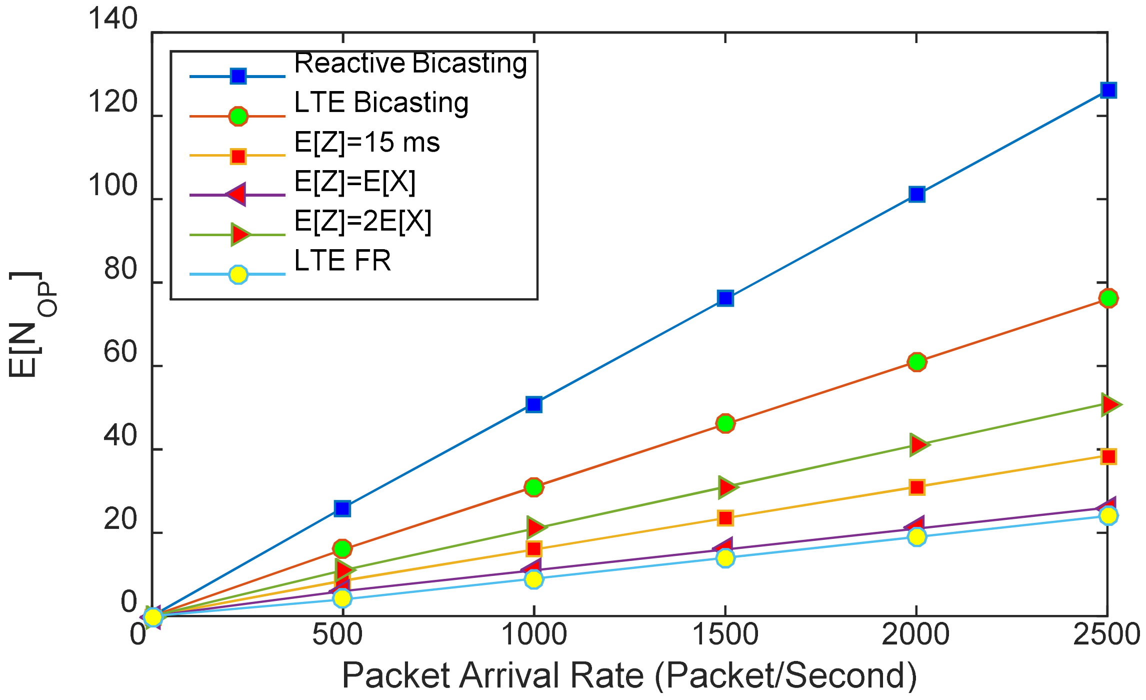


Figure 6.12: Effect of packet arrival rate on the expected overhead

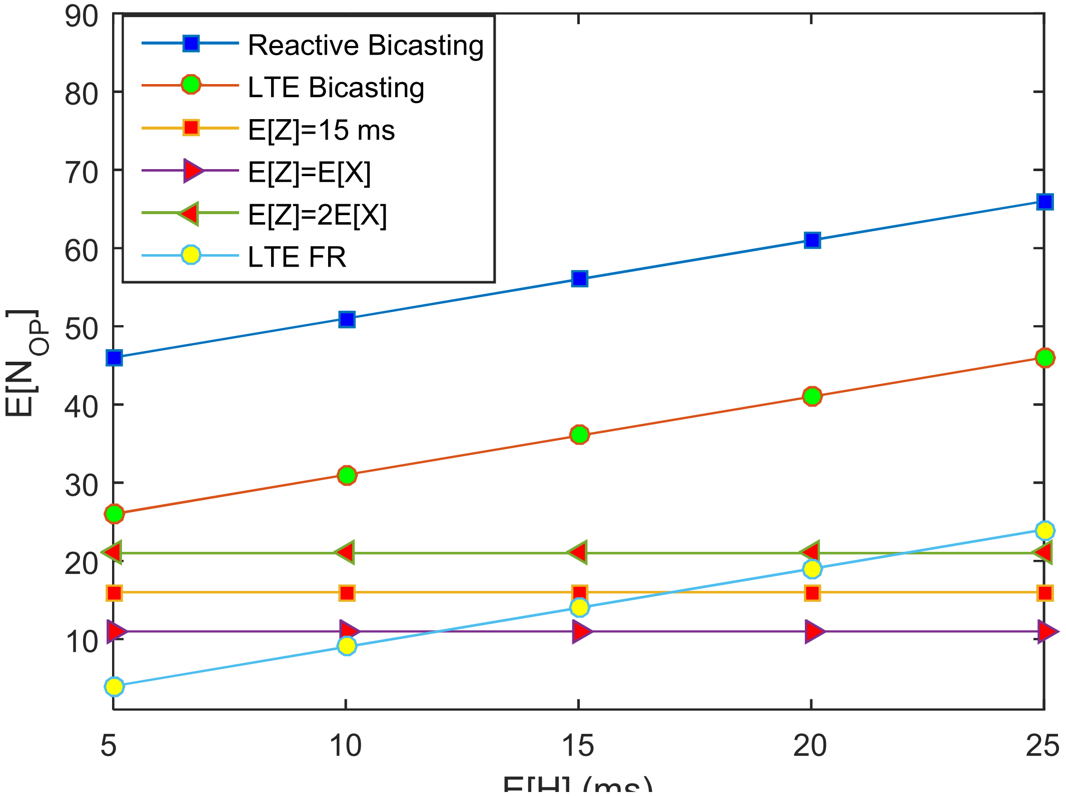
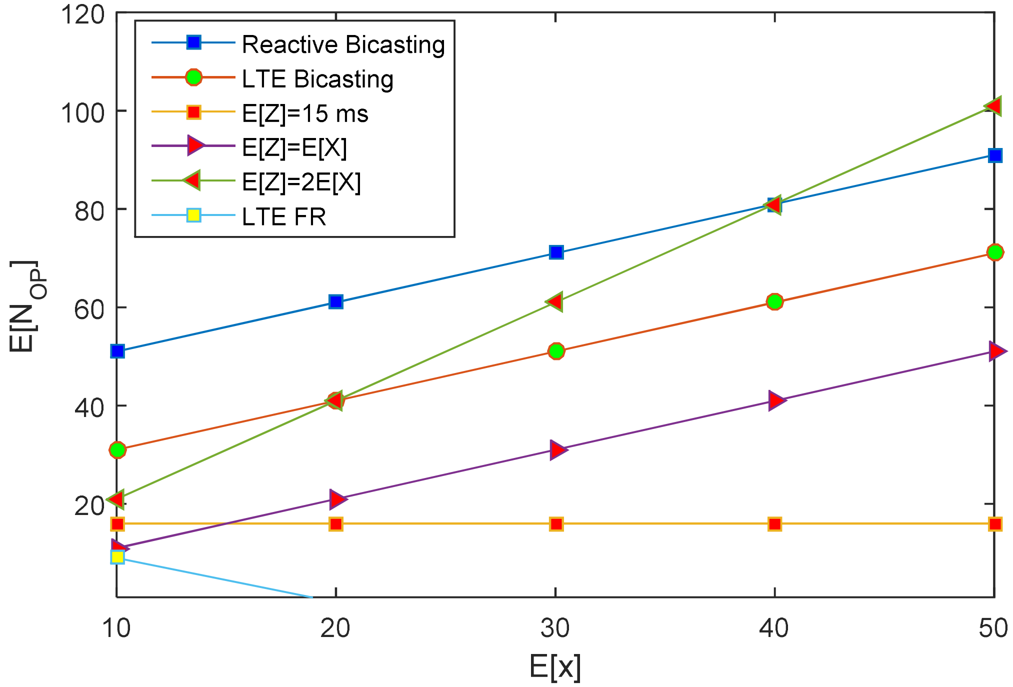


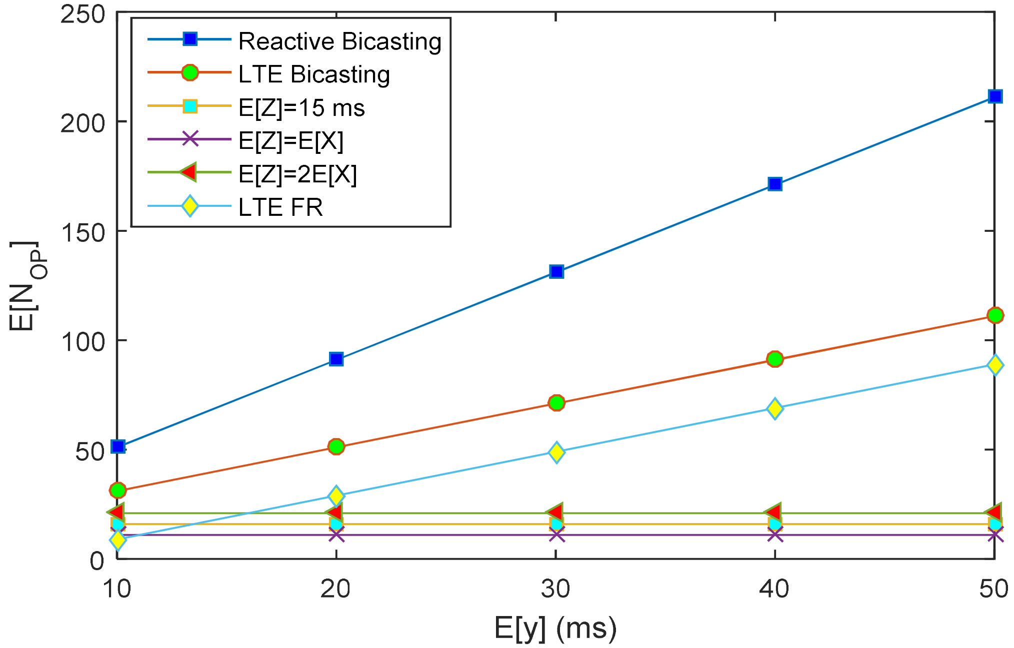
Figure 6.13: Effect of device handoff latency on the expected overhead

In Fig. 6.14 (a), we show the effect of increasing source latency on packet overhead. In the LTE Forwarding protocol the data is forwarded only if it reaches the source eNB, hence with the increase of source delay the packet overhead decreases. Using the bicasting period E[z] to be equal to double the source delay in the proposed protocol causes overhead to increases in a faster rate with the increase of source delay. However, we note that even using bicasting of double the source delay, it is still better than the Reactive Bicasting in most of cases. The best result after the LTE Forwarding is achieved using either constant bicasting period or a bicasting period equals to the source delay.

In Fig. 6.14 (b), we show the effect of the destination delay on the packet overhead. Reactive Bicasting and LTE Bicasting cause more overhead than the proposed protocols and the LTE Forwarding. The packet overhead caused by LTE Forwarding increases with the increase of the destination latency, due to the forwarding will not stop unless the *HO Notify* message is sent from target eNB and received successfully by the MME. However, for all the proposed protocol values, there is no effect on the overhead. Since the bicasting period is independent from the target delay.



(a) Effect of source latency



(b) Effect of destination latency

Figure 6.14: Effect of source and destination latencies on the expected overhead

In Fig. 6.15, we show the effect of relative network delay increase on the overhead. As expected the Reactive Bicasting is the highest in overhead and after it the LTE Bicasting. The proposed protocol with all the selected values performs better than both LTE Bicasting and Reactive Bicasting. However, LTE Forwarding does not change much with the change of the relative network delay, due to increasing both source and destination delay by the same percentage.

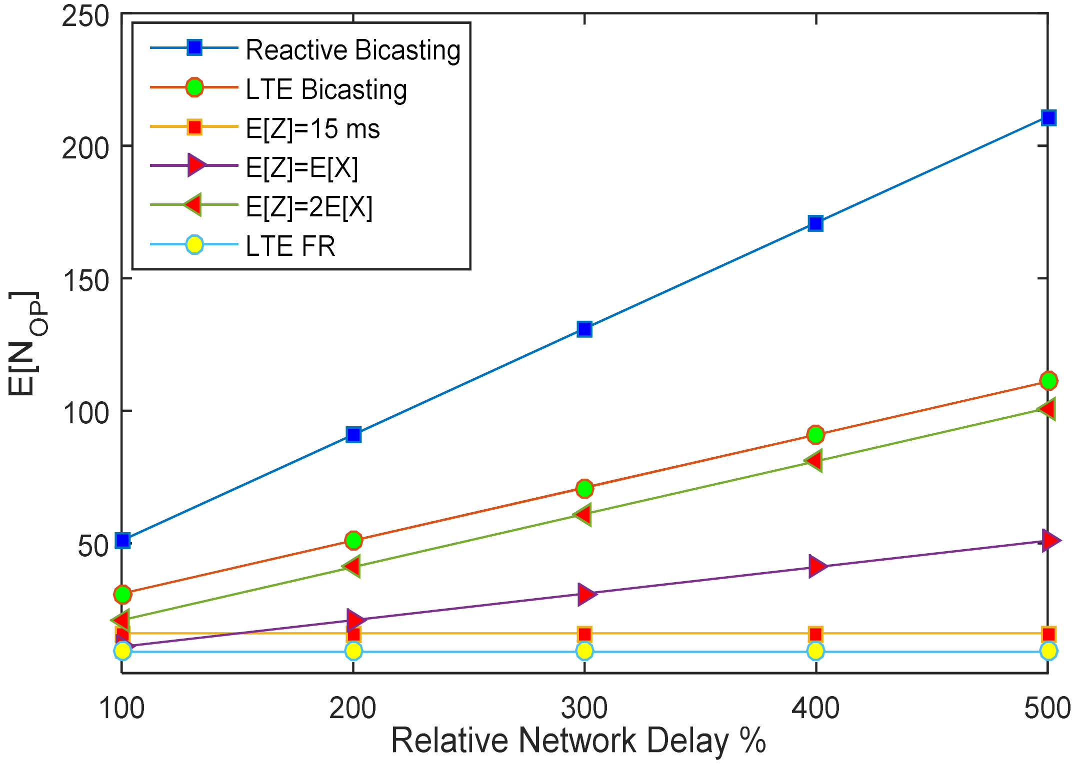


Figure 6.15: Effect of network delay increase on the expected overhead

6.6 Conclusion

Seamless mobility is a key issue in the next generation networks. There are several handover protocols have been proposed in the literature and by the 3GPP, specifically on the LTE networks and address the handoff between macrocells and femtocells. However, they are suffering from performance problems in terms of packet overhead, service distribution time, and packet loss. In this paper, we propose a self-organized reactive bicasting protocol for LTE networks. Its novelty lies in optimizing the start and the end of the bicasting to reduce packet overhead, in the same time take advantage of the bicasting to reduce the service disruption time. The proposed protocol is self-organized since it changes the bicasting time based on the source latency, destination latency, or both based on the operator QoS preferences. We provided a comprehensive analytical model for handover on LTE; we showed analytically that the proposed protocol could be adjusted to perform similarly to the standard LTE protocols of the state of the art one. Moreover, our goal in the paper is not to find the exact setting. However, it is to provide a flexible protocol that empowers network operators to enhance the handover experience based on their network preferences. The numerical results showed that advantages and disadvantages of the standard LTE protocols and the state of art one. Moreover, the results indicate that setting the proposed protocol's parameters lead to superior performance as compared to standard ones.

CHAPTER 7 METHOD FOR VEHICULAR WIRELESS COMMUNICATION WITH INCREASED RELIABILITY, QUALITY OF SERVICE QOS AND FEWER DISCONNECTIONS

7.1 Introduction

The innovation of this invention is a method to overcome the Dead spots by means of partner vehicles to ensure the maximum vehicle connectivity and the best wireless QoS. This invention also provides a wireless method to predict the dead spot region between two network coverage as well as within the same network coverage. In addition, this invention provides a pure vehicle to vehicle method to overcome the dead spot problems and perform the handoff as well as a network assisted method to do it. The current invention does not require participating vehicles to share their future routes neither with other vehicles nor with base stations or any other entity.

The vehicle industry is one of the pillars of North America`s economy. In 2011, more than 1.6 million new cars were sold in Canada with a total price of more than $54.1 billion [1]. In the recent years, there has been a significant jump in vehicular technologies; big companies are competing to provide more safety and entertainment applications and services to their customers. The vehicular network safety and infotainment industry is a relatively new and promising area of business.

In 2011, 68% of the internet users browsed the internet to watch and read news, and 74.1% of Canadians paid around $600 towards handheld devices and smartphones, in addition to $344 for internet services and $120 for online services [2]. Providing more infotainment services for vehicles are currently planned to be implemented in the future cars.

The present inventions relate to a method of wireless communication, more particularly, a method for vehicular wireless communication with increased reliability, Quality of Service QoS and fewer disconnections.

With the latest dramatic expansion in the vehicle electronics and wireless technology, advanced wireless applications are developed to suit the vehicular environment. Such applications include both safety and infotainment applications. Infotainment applications include Video-On-Demand, Voice over IP (VoIP), live streaming, news and weather broadcast, Internet Protocol Television

(IPTV) and traffic updates. Safety applications include smart lane change, road hazard warning, accident notifications, and cooperative collision avoidance

The quality of wireless signal between a vehicle and a wireless base station is highly affected by not only the distance of the vehicle to the base station but also the obstacles between a vehicle and the base station, for example, obstacles in the Fresnel zone. In the near ideal situations where there are no obstacles between the base station and the vehicle, the wireless signal is exponentially decaying compared to the distance between the base station and the vehicle. In the case of obstacles, such as building and tunnels, the signal is decaying depending on the obstacle material and its cross-sectional thickness.

Other contributions to handoff challenges are high dynamic mobility along with the unpredictability of driver behavior, vehicle density, and road traffic conditions. Also, due to the high speed of vehicles, the time available for proactive handoff between two networks is shortened and could be not enough for completing the handoff process. These contributors result in rapid and unexpected handoff between homogeneous and heterogeneous networks and network coverage disconnections which diminish QoS

The dead spot is the geographical areas where there is no or very poor signal strength, which jeopardize at least one of the vehicular wireless applications to work properly. Unpredicted loss of connectivity can cause the reconnection to the network afterward take longer time, loss of the reserved resources, and unpredicted state of the wireless resources booked for the vehicle in transit. Vehicles in the dead spot region are not able to use any of the described services which cause both disappointing infotainment user experience and disconnections of vital vehicle safety applications.

The innovation of this invention is a method to overcome the Dead spots using partner vehicles to ensure the maximum vehicle connectivity and the best wireless QoS, as shown in Fig. 7.1. This invention also provides a wireless method to predict the dead spot region between two network coverage as well as within the same network coverage. In addition, this invention provides a pure vehicle to vehicle method to overcome the dead spot problems and perform the handoff as well as a network assisted method to do it. The current invention does not require participating vehicles to share their future routes neither with other vehicles nor with base stations or any other entity.

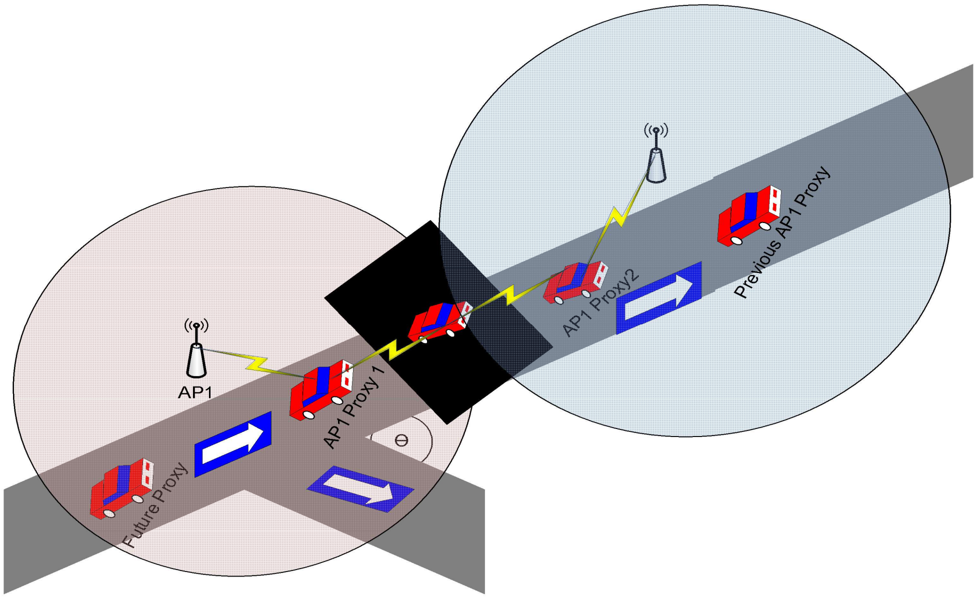


Figure 7.1: Dead Spot mitigation

7.2 Detailed Description

In this section, we describe in details the of the proposed invention.

1. A Wireless communication method contains the steps of: identifying the location of a dead spot in the current route within the current wireless coverage or in between the current wireless coverage and the next wireless network coverage, the vehicle should suffer from low or no signal strength detected; estimating the distance of the dead spot region to the current vehicle on the current route; estimating the distance from the vehicle till the next dead spot region on the route; and selecting at least one vehicle in the immediate wireless range to help in completing a delay sensitive and\or interactive video or audio streaming with at least one source outside the current vehicle wireless access; the interactive media content should be uninterrupted by the dead spot region and dependent upon the two estimating steps.

2. The method of claim 1, wherein the selection step includes: determining the respective time durations of a plurality of vehicles that are in the current wireless access of the vehicle; and at least one of these vehicles will be in both the wireless range of the current vehicle and the wireless range of the current access point coverage during the vehicle is passing through the dead spot region;

selecting the best vehicle that is expected to be in the current vehicle range; the selection has to be completed before the vehicle arrives at the dead spot region.

3. The method of claim 2 wherein the determining step includes: polling the vehicles in the direct wireless access with their current speed and location; estimating the time of each vehicle will be in the vehicle direct wireless range and the access point wireless range within and after the dead spot region.

4. The method of claim 2, wherein the determining step includes: sharing the current speed and location of the vehicle with the access point; polling the speed and location of all vehicles in the neighborhood of the current vehicle; estimating the time of each that will be in wireless range the current vehicle and will be have an acceptable wireless access during the period of time where current vehicle is passing through the dead spot region; and sending an ordered list of best vehicles to the current vehicle.

5. The method of claim 3 and claim 4 wherein the estimating step includes: using vehicles speed, direction, and location with the current dead spot region boundary and location and the road topology of the current road segment and\or the latest traffic information and\or the current and neighbor network coverage to calculate the duration of connectivity.

6. The method of claim 1 is compromising the further step of providing both the vehicle and all the partners in the interactive audio and/or video a warning with the expected disconnection time and the expected time of joining back the streaming. The vehicle end can take precautions of saving all the current sessions, and the other end can free and\or keep the session information until the vehicle joins back.

7.3 Comparative Analysis

In this section, we compare the proposed innovation to the state of the art patents. Our research in the patent database resulted in three patents that are closely related to the proposed one. In Table 7.1, we compare the proposed one to the selected patents. As shown in the comparison results, our proposed solves the problems of the patents, especially for the timing of when a vehicle will start the handoff process and its suitability for interactive applications.

Table 7.1: Patent Comparative Analysis

	US2010/0254346 A1 [110]	US2010/0240346A 1 [111]	US2011/0167128A 1 [112]	Proposed
Field of Patent	Proactive Handoff to Vehicular Environment	Dead Spot Prediction Method for wireless vehicular application	Dead Spot Mitigation Methods for Media Applications in Vehicular Environments	Proactive Handoff and proactive dead spot avoidance problem
V2I, V2V, or hybrid	V2I	V2I	V2I, Hybrid	All
Requires future route information of the vehicle to be shared?	Yes, security problem.	No	No	No
Using geographical static map information?	No	No	Yes	Yes
Require vehicle`s exact vehicle speed and position?	Yes	Yes	Yes	Optional

Table 7.2: Patent Comparative Analysis

Estimate the time before in between networks handoff?	Yes	Yes	Yes	Yes
Estimate the time that a vehicle will be in between handoff ?	Yes	Yes	Yes	Yes
Estimate the time when the vehicle have to start handoff?	No	No	No	Yes
Requires special hardware and/ or software on the vehicle?	Yes, maps, GPS, and an external server.	Yes, GPS, maps and external server	Yes, GPS, maps, sensors,	Digital maps on the base station or the vehicle
Dead spot location prediction?	No	Yes	No	Yes
Automatic Dead Spot network assisted avoidance?	No, only upon a vehicle request	No, only upon vehicle request	No	Yes
Dead spot avoidance in the same network?	No	Not clear, but generally Yes.	Yes	Yes

Table 7.1: Patent Comparative Analysis

Dead spot avoidance between two networks?	Yes	Yes	No	Yes
Require overlapping between all the networks?	Yes	No	No	No
Using partner vehicle	No	No	Yes, but not always	Yes
Optimizing the selection of partner vehicle?	No	No	Yes, based on the media content length	Yes, based on QoS and connectivity length
Is suitable for live and interactive application?	Yes	No	No	Yes
Ensuring connectivity?	Yes	No	No	Yes
Ensure the longest connectivity during and after Dead spot?	No	No	No	Yes

The usage of vehicles as access points has been discussed in Virtual Access Points (VAP) for Vehicular Networks [7]. The main difference between this paper and the proposed patent is that it does not use the vehicular topology knowledge to select a VAP, it does not optimize the selection of a VAP since it is the best effort, it does not estimation and eliminate known dead spots, and finally it does not enable vehicles to select partner vehicles, or partner VAPs, to help in the handoff.

The main similarities between the proposed and the “Dead Spot Prediction Method for wireless vehicular application” patent, is solving the problem of dead spots between networks and using time periods to describe dead spot regions and then take action. However, it is different in the way the problem is solved as follows:

1- The proposed patent is used for both interactive and offline streaming. However, the other one is only suitable for offline streaming, since it uses the media length to estimate the time required to finish streaming.

2- In the mentioned patent there is not network assisted handoff that requires no intervention for vehicles on the road that contains GPS. The proposed patent proposes a mode where the network can detect a vehicle that will pass through the handoff, and then assign a partner vehicle to be used as a relay to avoid disconnections. In this mode having GPS devices is optional and the network can calculate the average speed and select the right partner seamlessly.

3- The mentioned patent disconnects at the time of the handoff (i.e. there is not data exchange during the dead spot), so the handoff problem is neither avoided nor reduced.

4- The proposed patent is using partner vehicle to avoid the disconnection during the handoff.

5- The mentioned only detect the location of the dead spots without a mitigation technique. By knowing the exact location using a GPS as mentioned in the patent“Current GPS coordinates and the vehicle route.” However, there is no specific algorithm and protocol to specify what the action that should be taken is.

6- The mentioned patent requires an external server to store the dead spot location. Unlike the proposed one where a real time v2v can be used to detect a dead spot and avoid it.

7- The rest of the mentioned patent is totally different from the proposed one, since it does not deal with using a partner, selecting the best partner, etc.

7.4 Solution illustration

To avoid dead spots, the network has to first detect their relative locations to advice roaming nodes about the approximate locations of the dead spots. In Fig. 7.2, we show our proposed method to detect and enhance the location of a dead spot. Since the access point and base station have the ability to detect link interruption, we are taking advantage of this property by keeping track of these triggers to estimate the start of the dead spot boundary. The network is capable of estimating the vehicle speed, and hence the relative location of the start of the dead spot boundary can be calculated. Similarly in Fig. 7.3, we show a method to detect the estimated location of a dead spot, by tracking the link connection after a vehicle leaves the dead spot zone.

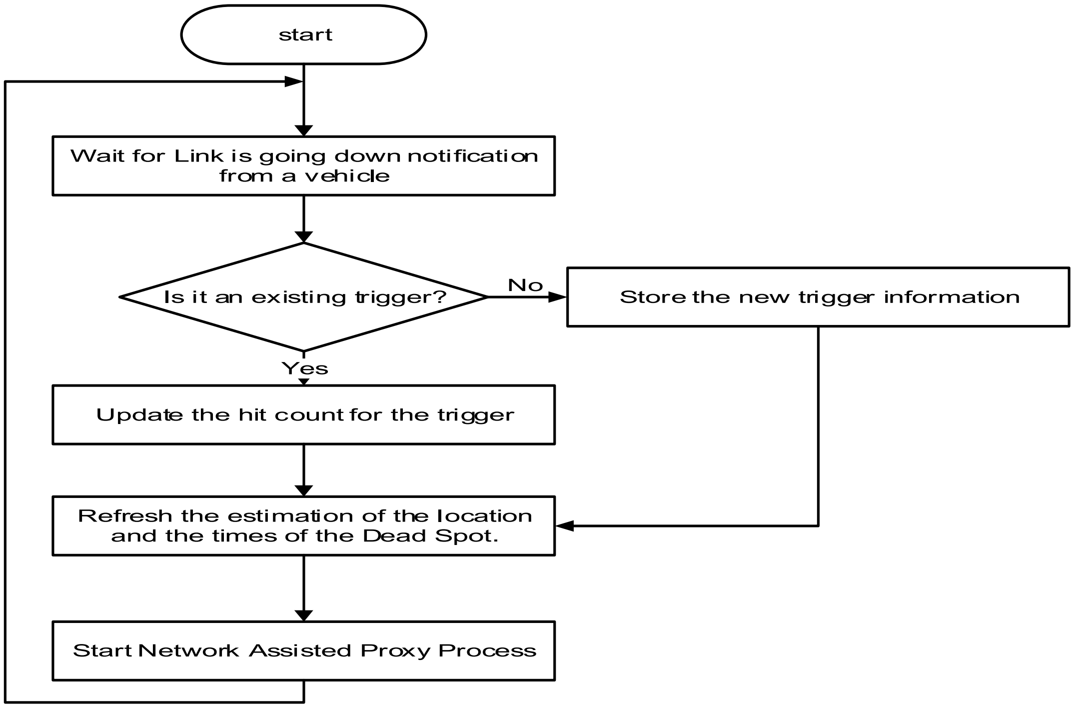


Figure 7.2: A method to detect the start of a dead spot boundary

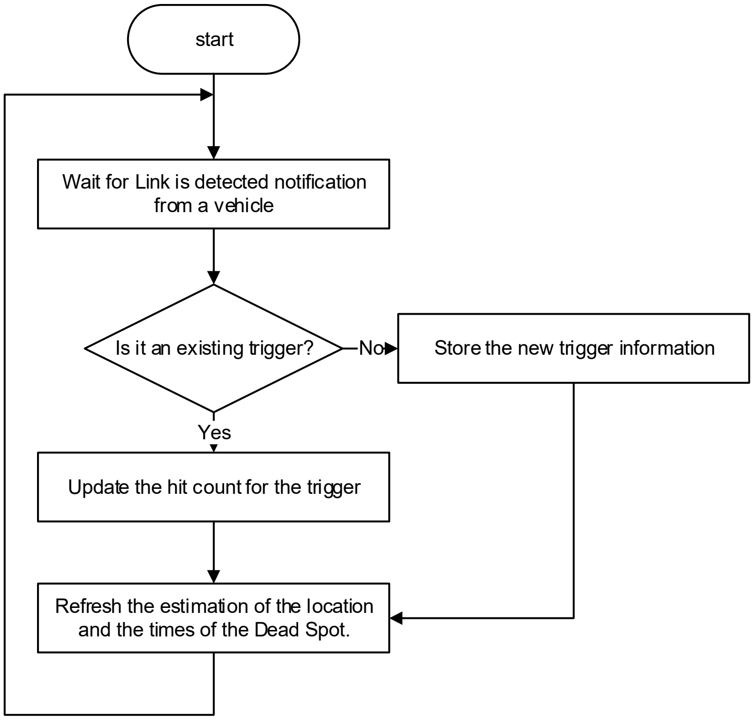


Figure 7.3: A method to detect the end of a dead spot boundary

The proposed solution provides a protocol to inform roaming vehicles about dead spot locations and the time time to connect to a proxy as shown in Fig. 7.4. Using the vehicle approximate position, that can be found using the signal strength and network fingerprinting, the base station evaluates calculates the expected time when the QoS will start degrading due the the dead spot (T_a), the expected time when the vehicle will face the dead spot (T_b) and must connect to a proxy, the expected time that the vehicle will exit the dead spot (T_c), and the expected time that the vehicle will exit the current network coverage and must connect to the new network (T_d).

In Fig. 7.5, we show a network assisted proxy assignment method. In this method, the base station is keeping track of each vehicle, and whenever the base station predicted that a vehicle is suitable to work as a proxy, it shares with it a security key that is used only for delivering, or relying on, the information from partner nodes. Afterward, the network advertises this vehicle as a proxy. Since the current proxy will be approaching the dead spot boundary soon, the base station starts searching for a new proxy in the proximity of the current proxy.

In Fig. 7.6, we present a method to calculate and estimated T_a, T_b, T_c, T_d. As shown, the method is making use of maps and traffic to expect all the route probabilities for the roaming vehicle. Moreover, if the location is not available, the base station will try to estimate the location using the neighbor list, and if there are no neighbours detected it would use the expected average speed along with the last time it left the last network.

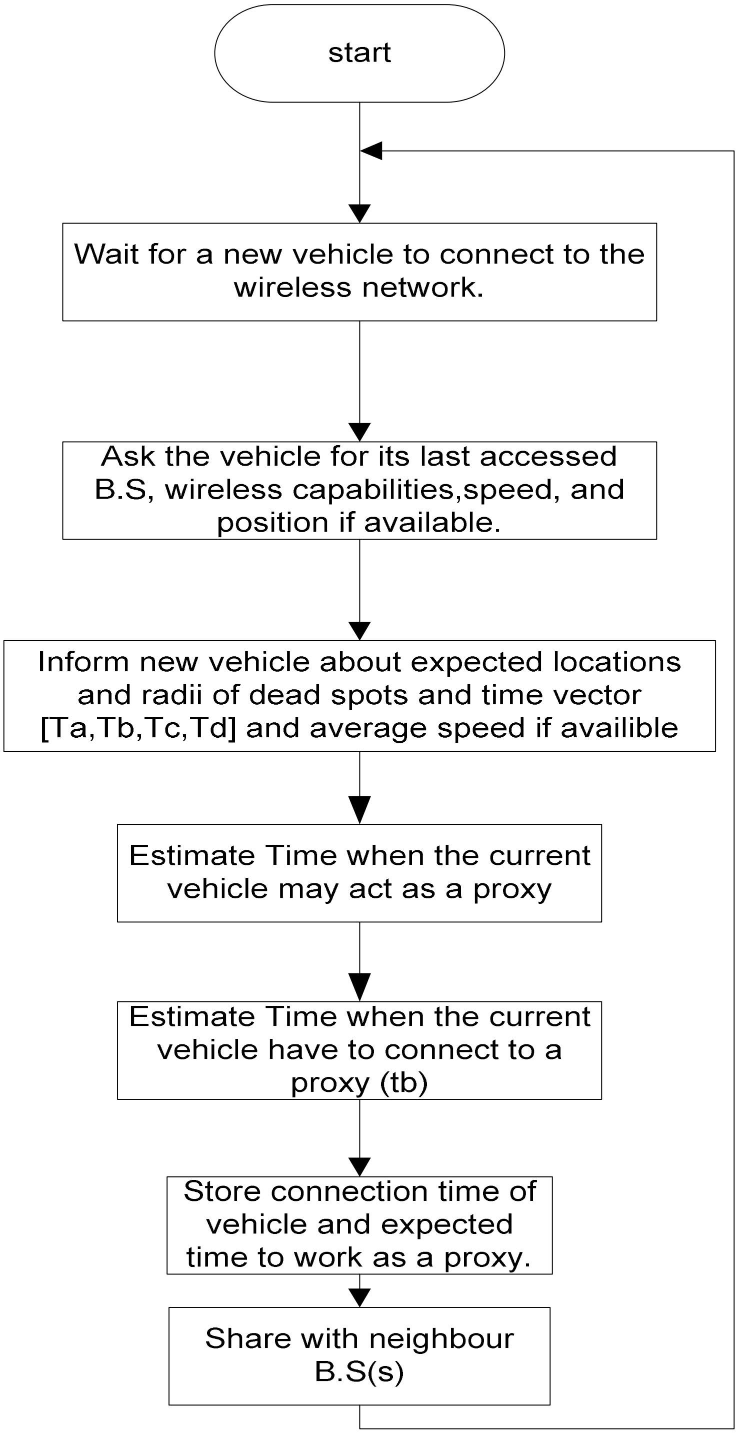


Figure 7.4: Estimating dead spot location for a new vehicle

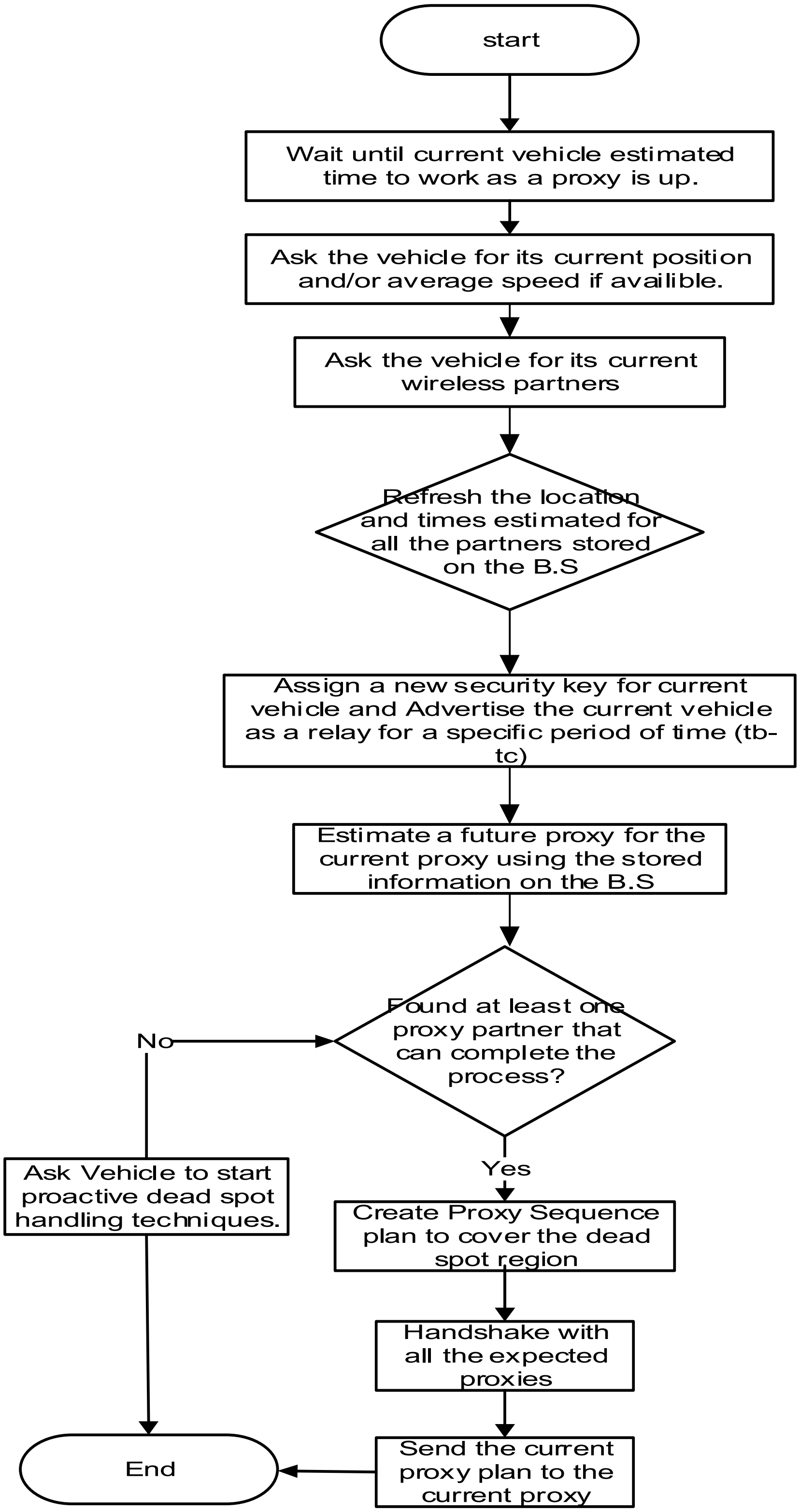


Figure 7.5: Network assisted proxy assignment

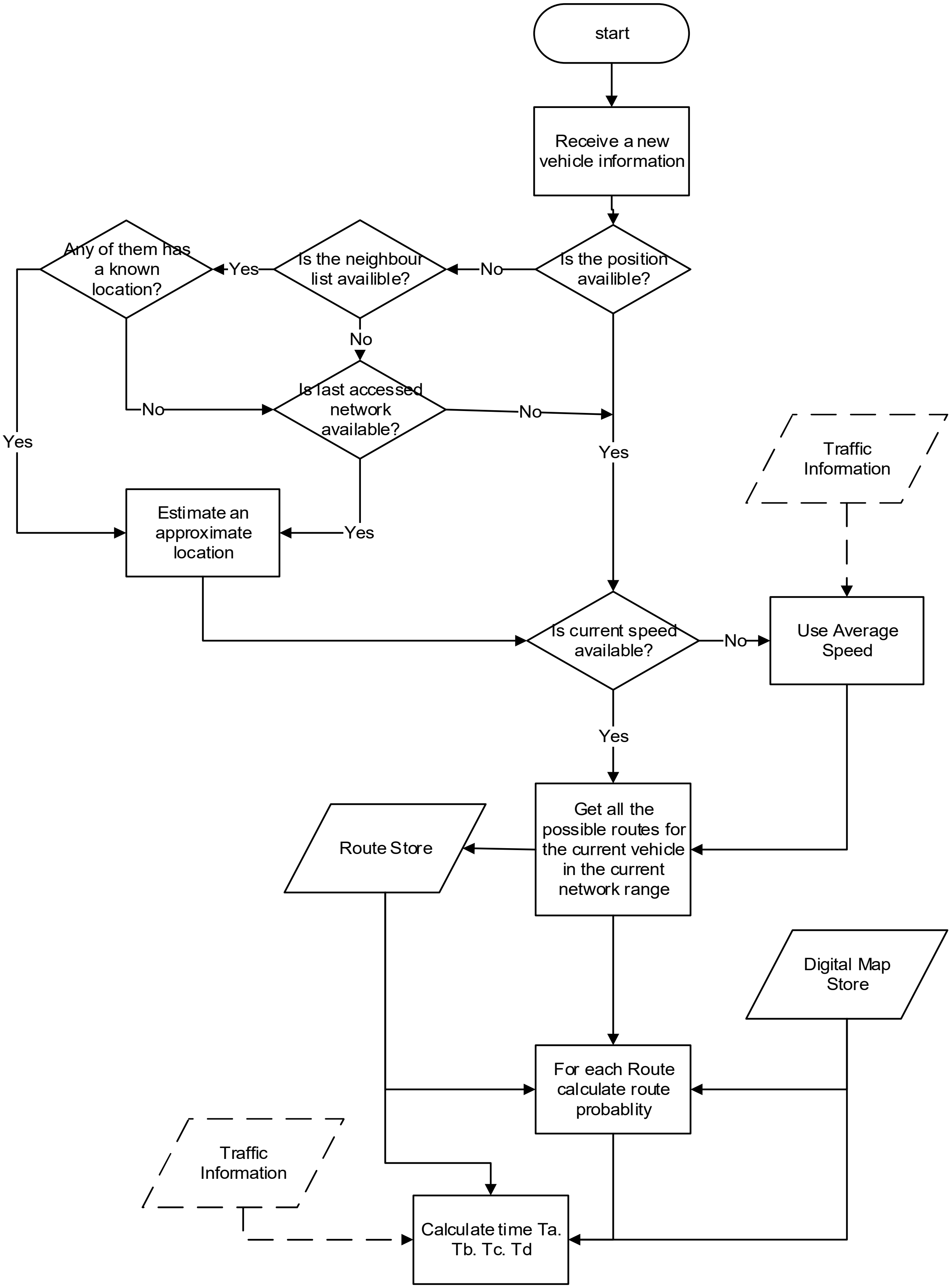


Figure 7.6: A method to calculate routes and estimate dead spot boundaries

7.5 Conclusion

In this chapter, we presented an innovative way to detect and avoid dead spots. We held a comparative analysis to analyze the similarities and differences between the proposed protocol and the state of the art patents. The analysis shows that the presented protocol and methods are unique and solving the problems of the compared patents. Moreover, we demonstrated the core methods of our proposal to detect and mitigate the dead spot problems on a wireless network.

CHAPTER 8 GENERAL DISCUSSION

The work carried out in this thesis, introduces server novel protocols, schemes, architectures, and prototypes for vehicular networks. In general, the main contribution of this thesis is focused on solving the handoff problems in VANET. We traced the roots of handoff problems on VANET and provided innovative yet practical solutions that effectively enhance the handoff on VANET. We attacked the handoff problems from several angles such as infrastructure tailored, innovative network side, novel partner side, and LTE-based handoff protocols. Moreover, we attacked the lack of testbed for VANET handoff protocol, by providing an architecture and a prototype VANET testbed.

New wireless infrastructure is based on fiber optics, specifically PON, instead of the common copper wires. PON provides excellent bandwidth along with low delay that is essential for media applications. However, the traditional handoff protocols are not designed to make benefit of the PON to enhance the handoff process, as explained in chapter 2.

PON broadcasts all the received data to the ONUs where each ONU will only process its data and drop the rest. In chapter 2, we employed this feature in reducing the handoff packet overhead that is caused by traditional forwarding techniques.

We believe that the key to developing practical VANET applications and protocols lies in providing an efficient emulation and simulation tools. Chapter 3, provides a comprehensive analysis for network and mobility simulators and emulators that can be used for VANET. It provides the advantages, popularity, and recommendations for each tool and in which context it can be used.

Traditional wireless applications are tested using network simulators. However, those lack the mobility aspects of VANET. The integrations between network simulators and mobility simulators have risen as an effective tool to simulate vehicular applications and protocols. However, as mentioned above, no suitable testbed can be easily used to emulate VANET applications and protocols. Chapter 4, presents a flexible testbed architecture that can be used to emulate vehicular applications and protocols effectively. The proposed architecture integrates network and mobility simulators, along with sensor virtualization to emulate VANET applications effectively.

Chapter 5, presents a partner assisted handoff protocol that is tailored for VANET. In this protocol, roaming vehicles can search for partner vehicle to use during the handoff to stay connected to the network. The partner assisted handoff has shown excellent potential in solving handoff problems that traditional network based handoff protocols failed to solve. However, selecting the right partner promptly is the most important factor in the partner assisted handoff success. Hence, in Chapter 5 we focus on an effective partner selection protocol that predicts all the routes of partner vehicles to determine the one that is expected to stay longer in the range. Also, the proposed protocol make use of the static maps and average road speed, at the same time, does not require partners' routes.

Chapter 6, studies in depth the handoff for LTE networks, with a focus on the handling of handoff among networks with different Quality of Service (QoS) requirements such as delay. With the emerge of LTE-A networks, it is expected that the handoff will be become more challenging for vehicular networks due to rapid handoff between networks resulting in shorter time available for the handoff to be completed.

In Chapter 6, we introduced a novel protocol for LTE handoff that is suitable for femtocell and macrocell scenario. The proposed protocol is compared against the state of the art protocols, and the results show the superiority of the proposed one regarding QoS. Moreover, we derived an analytical model, with its mathematical proof, that is used to compare the LTE handoff protocols and can be employed in the future by other researchers to evaluate the next generation handoff protocols.

One of the handoff reasons is the degradation of QoS in the network boundary. Weather conditions along with other obstacles on the boundary of a wireless network coverage can create small areas with limited or no connectivity. These small areas are called “Dead Spots.” When a car suffers from a degradation of a wireless QoS by approaching a dead spot, it may start handoff prematurely to the next network, and then handoff back to the original network. This phenomenon is known as the ping-pong effect. Chapter 7 provides a solution for the dead spot problems in VANET. Not only it informs the vehicles about the locations of the dead spots but also assigns a partner vehicle to work as a proxy or a relay to keep the vehicle connected while it is in the dead spot.

CHAPTER 9 CONCLUSION AND RECOMMENDATIONS

The purpose of this chapter is to highlight the contributions, limitations, and future work of the research that is explained in this thesis.

9.1 Research Contribution

This thesis tackled the problem of handoff on the vehicular networks from multiple angles.

1- We introduced a novel protocol for reducing packet overhead results from the handoff. Also, we introduce an efficient integration between PMIP mobility management protocol and the PON infrastructure.

2- We conducted in-depth survey for all simulators, emulators, and testbeds that are currently in the market. We analyzed the points of weakness and strength in each of them and provided recommendations for the vehicular testbed. This research is valuable for being an excellent guide for any researcher that will be working on VANET using simulation or emulation or will be working on enhancing the current emulators and simulators.

3- We presented and prototyped a flexible testbed architecture for vehicular networks that can be used to verify and validate vehicular applications. The suggested architecture is based on open source platforms and can be utilized along with any simulator in the market. Our contribution to this research helps faster and more accurate adoption of vehicular applications.

4- We introduced a handoff protocol that is tailored for VANETs; it utilized partner vehicles to help in the handoff process. The innovation of this protocol is in selecting the right partner; we used road topology to predict partner's movements and choose the best based on the matching of partners' expected routes with the roaming vehicle one. To our knowledge, there no other handoff protocol that is selecting best partners based on the prediction of their mobility.

5- We introduced Vehicular Link Expiry Time (VLET) that is measuring the link expiry between two vehicles without sharing the expected route. On the contrary of the mainstream research that requires partner's route to predict the Link Expiry Time (LET) or using random mobility and direction, VLET is operating without the need for the partner's route.

Hence, it is practical and more secure. Performance analysis shows the superiority of VLET in selecting partners as compared to standard and state of the art partner selection protocols.

6- We introduced an innovative network-based bicasting mobility management protocol. The innovation of this protocol that does not depend mainly on the handoff signaling to start and stop bicasting, however, it depends on the delay of the source and destination networks to optimize the bicasting. Moreover, the introduced protocol gives the network operators the ability to enhance the QoS by balancing packet loss to handoff delay based on their network requirements.

7- We formulated and derived a closed-form equation for the packet overhead on LTE networks by using Mixed Erlang functions. Along, with the other derived closed form equations, we created an analytical framework that can be employed by any researcher to analyze other innovative LTE handoff protocols.

9.2 Research Limitations

Despite the contributions, stated above, the work in this thesis has certain limitations.

1- The proposed PMIP and PON integration is optimizing the broadcasting of data by using the native downstream broadcasting properties of the PON. However, using this architecture in a delay sensitive environment requires different sleep cycles for the ONUs to enable the packet inspection of incoming data frames.

2- The proposed flexible testbed is using multi-stage emulation concept, where the network and mobility simulations are carried out in one stage, then the emulation is using the simulation results to extract KPI and apply them to the communication channel. Although this method is proven to produce practical results, using KPIs in scenarios that are completely different than the simulation may lead to synthetic data.

3- Although the proposed flexible testbed architecture can be used to emulate vehicular applications on any operating system, it requires modification in the operating system itself to emulate all sensors. Although Android is an open source, other operating systems such as Apple's IOS is not, which limits the testbed to be implemented on Android and other open source operating systems.

4- The proposed partner based mobility management protocol relies heavily on partner vehicles to assist in the handoff. Hence, it requires an adoption of the protocol by several cars manufactures to have the benefit of the cooperation.

5- We derived a closed-form equation for the packet overhead using mixed Erlang random variables for LTE networks. However, to provide a similar equation for buffering that can be used for forwarding based protocols, this requires dealing with the subtraction of multiple mixed Erlang random variables which results in complex numerical equations that don't have a simple closed form.

9.3 Direction for Future Work

At this stage, there are several venues for future research. Some of the future research lies in the limitations stated above, applications for the developed algorithms, and enhancements to the proposed protocols.

For the handoff testbed, we will investigate more possibilities of using a feedback simulation to the emulation platform as opposed to the multi-stage approach. Although it is more complex to used feedback systems, it can be integrated with the multistage approach to creating near to real life scenarios experience. We may also have the opportunity to implement the proposed architecture on a larger scale and verify the performance on large scale emulations.

The proposed road topology based partner assisted handoff protocol can be modified to work to rely more on the network rather than the partner vehicle. We are interested to in investigating the ability of base stations to predict node movements based on the times it got detected entering and exiting from wireless coverage zones.

The proposed VLET metric can be used in Ad-hoc routing as well. To our knowledge, no ad-hoc routing algorithm makes use of partners' route prediction to select the next hop. However, they tend to share the partners' route which causes a security challenge.

Finally, the proposed self-organized bicasting handoff protocol will be analyzed using network and mobility simulation on vehicular, and the results will be compared to the mathematical model. The aim of this research is to study the effect of mobility and driver behavior on the derived mathematical model.

BIBLIOGRAPHY

[1] L. T. E. Motorola, "A technical overview," *Technical White Paper,* 2007.

[2] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *Intelligent Transportation Systems Magazine, IEEE,* vol. 6, pp. 6-22, 2014.

[3] L. Jorguseski, A. Pais, F. Gunnarsson, A. Centonza, and C. Willcock, "Self-organizing networks in 3GPP: Standardization and future trends," *Communications Magazine, IEEE,* vol. 52, pp. 28-34, 2014.

[4] P. Guturu, "Explosive Wireless Consumer Demand for Network Bandwidth-Fifth Generation and Beyond [Future Directions]," *IEEE Consumer Electronics Magazine,* vol. 6, pp. 27-31, 2017.

[5] M. Maier, M. Levesque, and L. Ivanescu, "NG-PONs 1&2 and beyond: the dawn of the uber-FiWi network," *Network, IEEE,* vol. 26, pp. 15-21, 2012.

[6] S. Winkler, C. W. Chen, A. Raake, P. Schelkens, and L. Skorin-Kapov, "Introduction to the Special Issue on Measuring Quality of Experience for Advanced Media Technologies and Services," *IEEE Journal of Selected Topics in Signal Processing,* vol. 11, pp. 3-5, 2017.

[7] I. Ali, A. Casati, K. Chowdhury, K. Nishida, E. Parsons, S. Schmid*, et al.*, "Network-based mobility management in the evolved 3GPP core network," *Communications Magazine, IEEE,* vol. 47, pp. 58-66, 2009.

[8] M.-S. Kim, S. Lee, D. Cypher, and N. Golmie, "Performance analysis of fast handover for proxy mobile IPv6," *Information Sciences,* 2012.

[9] V. Sandonis, M. Calderon, I. Soto, and C. J. Bernardos, "Design and performance evaluation of a PMIPv6 solution fro geonetworking-based VANETs," *Ad Hoc Networks,* 2012.

[10] J.-H. Lee, T. Ernst, and N. Chilamkurti, "Performance analysis of PMIPv6-based network mobility for intelligent transportation systems," *Vehicular Technology, IEEE Transactions on,* vol. 61, pp. 74-85, 2012.

[11] T. Guo, N. Wang, and R. Tafazolli, "Local mobility management for networked femtocells based on X2 traffic forwarding," *Vehicular Technology, IEEE Transactions on,* vol. 62, pp. 326-340, 2013.

[12] R.-H. Liou, Y.-B. Lin, and S.-C. Tsai, "An investigation on LTE mobility management," *Mobile Computing, IEEE Transactions on,* vol. 12, pp. 166-176, 2013.

[13] L. Yan, X. Fang, and Y. Fang, "Control and data signaling decoupled architecture for railway wireless networks," *Wireless Communications, IEEE,* vol. 22, pp. 103-111, 2015.

[14] K. Dongwook, L. Hanjin, and K. Namgi, "A velocity-based bicasting handover scheme for 4G mobile systems," *IEICE transactions on communications,* vol. 92, pp. 288-295, 2009.

[15] T. Guo, A. U. Quddus, and R. Tafazolli, "Seamless handover for LTE macro-femto networks based on reactive data bicasting," *Communications Letters, IEEE,* vol. 16, pp. 1788-1791, 2012.

[16] Y.-S. Chen and W. Kun-Lin, "A cross-layer partner-assisted handoff scheme for hierarchical mobile IPv6 in IEEE 802.16e systems," *Wireless Communications and Mobile Computing,* vol. 11, pp. 522-541, 2011.

[17] Y.-S. Chen and J.-S. Hong, "A relay-assisted protocol for spectrum mobility and handover in cognitive LTE networks," *Systems Journal, IEEE,* vol. 7, pp. 77-91, 2013.

[18] T. Taleb and K. Ben Letaief, "A Cooperative Diversity Based Handoff Management Scheme," *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS,* vol. 9, April 2010.

[19] A. Awada, B. Wegmann, I. Viering, and A. Klein, "A SON-based algorithm for the optimization of inter-RAT handover parameters," *Vehicular Technology, IEEE Transactions on,* vol. 62, pp. 1906-1923, 2013.

[20] T. Ueda, "Arrayed waveguide grating." U.S. Patent 6,498,878, issued December 24, 2002.

[21] F. Effenberger, D. Clearly, O. Haran, G. Kramer, R. D. Li, M. Oron*, et al.*, "An introduction to PON technologies [Topics in Optical Communications]," *Communications Magazine, IEEE,* vol. 45, pp. S17-S25, 2007.

[22] N. Madamopoulos, S. Peiris, N. Antoniades, D. Richards, B. Pathak, G. Ellinas*, et al.*, "A fully distributed 10G-EPON-based converged fixed-mobile networking transport infrastructure for next generation broadband access," *Optical Communications and Networking, IEEE/OSA Journal of,* vol. 4, pp. 366-377, 2012.

[23] A. Seddik-Ghaleb, Y. Ghamri-Doudane, and S.-M. Senouci, "Emulating End-to-End Losses and Delays for Ad Hoc Networks," in *ICC*, 2007, pp. 3224-3231.

[24] J. Garcia, P. Hurtig, and A. Brunstrom, "KauNet: A versatile and flexible emulation system," *Proceedings of SNCNW,* 2008.

[25] R. Lubke, P. Buschel, D. Schuster, and A. Schill, "Measuring accuracy and performance of network emulators," in *Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on*, 2014, pp. 63-65.

[26] M. Carson and D. Santay, "NIST Net: a Linux-based network emulation tool," *SIGCOMM Comput. Commun. Rev.,* vol. 33, pp. 111-126, 2003.

[27] S. Hemminger, "Network emulation with NetEm," in *Linux Conf Au*, 2005, pp. 18-23.

[28] L. Nussbaum and O. Richard, "A comparative study of network link emulators," in *Proceedings of the 2009 Spring Simulation Multiconference*, 2009, p. 85.

[29] E. Conchon, T. Pérennou, J. Garcia, and M. Diaz, "W-NINE: a two-stage emulation platform for mobile and wireless systems," *EURASIP Journal on Wireless Communications and Networking,* vol. 2010, p. 3, 2010.

[30] B. Lin, J. Tapolcai, and P.-H. Ho, "Dimensioning and Site Planning of Integrated PON and Wireless Cooperative Networks for Fixed Mobile Convergence," *Vehicular Technology, IEEE Transactions on,* vol. 60, pp. 4528-4538, 2011.

[31] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213 (Proposed Standard)August 2008.

[32] R. Roy, G. Kramer, M. Hajduczenia, and H. J. Silva, "Performance of 10G-EPON," *Communications Magazine, IEEE,* vol. 49, pp. 78-85, 2011.

[33] R. S. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. W. Ayre, and W. V. Sorin, "Evolution of WDM optical IP networks: A cost and energy perspective," *Journal of Lightwave Technology,* vol. 27, pp. 243-252, 2009.

[34] N. Ghazisaidi and M. Maier, "Hierarchical frame aggregation techniques for hybrid fiber-wireless access networks," *Communications Magazine, IEEE,* vol. 49, pp. 64-73, 2011.

[35] G. Koodli. (July 2005). *Fast handovers for mobile IPv6*.

[36] H. Yokota, S. Gundavelli, X. Cui, and J. Korhonen, "Runtime Local Mobility Anchor (LMA) Assignment Support for Proxy Mobile IPv6," in *RFC*, ed, 2012.

[37] I. Hwang, A. Nikoukar, C.-H. Teng, and K. R. Lai, "Scalable Architecture for VOD Service Enhancement Based on a Cache Scheme in an Ethernet Passive Optical Network," *Journal of Optical Communications and Networking,* vol. 5, pp. 271-282, 2013.

[38] (December, 2015). *The Network Simulator - ns-2*. Available: http://www.isi.edu/nsnam/ns/ (accessed: December, 2015)

[39] OMNET++ Module for LTE simulte [Online]. Available: https://github.com/inet-framework/simulte (accessed: December, 2015)

[40] *SUMO – Simulation of Urban MObility*. Available: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/

[41] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks," in *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*, 1998, pp. 154-161.

[42] QualNet [Online]. Available: http://web.scalable-networks.com/content/qualnet (accessed: December, 2015)

[43] A. Sobeih, J. C. Hou, L.-C. Kung, N. Li, H. Zhang, W.-P. Chen*, et al.*, "J-Sim: a simulation and emulation environment for wireless sensor networks," *Wireless Communications, IEEE,* vol. 13, pp. 104-119, 2006.

[44] *J-Sim Official*. Available: https://sites.google.com/site/jsimofficial/ (accessed: December, 2015)

[45] *NS2 LTE Model*. Available: https://code.google.com/p/lte-model/ (accessed: December, 2015)

[46] K. Renard, C. Peri, and J. Clarke, "A performance and scalability evaluation of the ns-3 distributed scheduler," in *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, 2012, pp. 378-382.

[47] A. P. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Real-world environment models for mobile network evaluation," *Selected Areas in Communications, IEEE Journal on,* vol. 23, pp. 622-632, 2005.

[48] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: generating realistic mobility patterns for VANETs," in *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, 2006, pp. 96-97.

[49] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," in *Fundamentals of Traffic Simulation*, ed: Springer, 2010, pp. 63-93.

[50] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: realistic joint traffic and network simulator for VANETs," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 12, pp. 31-33, 2008.

[51] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *Mobile Computing, IEEE Transactions on,* vol. 10, pp. 3-15, 2011.

[52] M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi*, et al.*, "ITETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications," *Simulation Modelling Practice and Theory,* vol. 34, pp. 99-125, 2013.

[53] D. Herrscher and K. Rothermel, "A dynamic network scenario emulation tool," in *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*, 2002, pp. 262-267.

[54] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo*, et al.*, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Wireless Communications and Networking Conference, 2005 IEEE*, 2005, pp. 1664-1669.

[55] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak*, et al.*, "Planetlab: an overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review,* vol. 33, pp. 3-12, 2003.

[56] L. Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," *ACM SIGCOMM Computer Communication Review,* vol. 27, pp. 31-41, 1997.

[57] J. Uddin and J. M. Kim, "Impact of Background Utilization and Background Traffic on the Foreground Applications in a Wide Area Network," in *Computer Science and its Applications*, ed: Springer, 2012, pp. 903-916.

[58] D. Kotz and T. Henderson, "Crawdad: A community resource for archiving wireless data at dartmouth," *Pervasive Computing, IEEE,* vol. 4, pp. 12-14, 2005.

[59] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *Mobile Computing, IEEE Transactions on,* vol. 13, pp. 1061-1075, 2014.

[60] C. Phillips, D. Sicker, D. Grunwald, and S. Singh, "Techniques for simulation of realistic infrastructure wireless network traffic," in *Modeling and Optimization in Mobile, Ad Hoc,*

and Wireless Networks, 2009. WiOPT 2009. 7th International Symposium on, 2009, pp. 1-7.

[61] P. Merino and A. Salmerón, "Combining SPIN with ns-2 for Protocol Optimization," in *Model Checking Software*, ed: Springer, 2010, pp. 40-57.

[62] A. Diaz, P. Merino, and A. Salmeron, "Obtaining Models for Realistic Mobile Network Simulations using Real Traces," *IEEE Communications Letters,* vol. 15, pp. .782-784, 2011.

[63] (2014, August). Available: https://www.google.com/events/io

[64] NewZoo, "AppLift And Newzoo Expect Mobile Games Market To Double In Size Until 2016 And Reach $23.9BN, Europe Fastest Growing Market," ed. Online, 2013.

[65] (2017, January). *Smart Device Link*. Available: https://smartdevicelink.com/consortium/

[66] A. Research. (2012). *$200 Million Mobile Application Testing Market Boosted by Growing Demand for Automation*. Available: https://www.abiresearch.com/press/200-million-mobile-application-testing-market-boos

[67] N. Aschenbruck, A. Munjal, and T. Camp, "Trace-based mobility modeling for multi-hop wireless networks," *Computer Communications,* vol. 34, pp. 704-714, 2011.

[68] A. Alvarez, R. Orea, S. Cabrero, X. G. Pañeda, R. García, and D. Melendi, "Limitations of network emulation with single-machine and distributed ns-3," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 2010, p. 67.

[69] J. Chon and H. Cha, "LifeMap: A Smartphone-Based Context Provider for Location-Based Services," *IEEE Pervasive Computing,* vol. 10, pp. 58-67, 2011.

[70] H. Yoon, J. Kim, and R. Hsieh, "Peer-assisted video on-demand streaming system in practical WiFi-based mobile opportunistic networks," *Journal of Network and Computer Applications,* vol. 37, pp. 33-44, 2014.

[71] V. Kasapakis and D. Gavalas, "Pervasive gaming: Status, trends and design principles," *Journal of Network and Computer Applications,* vol. 55, pp. 213-236, 2015.

[72] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Adaptive deployment and configuration for mobile augmented reality in the cloudlet," *Journal of Network and Computer Applications,* vol. 41, pp. 206-216, 2014.

[73] C. di Flora, M. Ficco, S. Russo, and V. Vecchio, "Indoor and outdoor location based services for portable wireless devices," in *The 25th IEEE International Conference on*

Distributed Computing Systems Workshops (ICDCSW'05), Ohio, USA,, June 2005, pp. 244- 250.

[74] P. Eugster, B. Garbinato, and A. Holzer, "Pervaho: A Development & Test Platform for Mobile Ad hoc Applications," in *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, July 2006.

[75] Y. Chen, X. Y. Chen, F. Y. Rao, X. L. Yu, Y. Li, and D. Liu, "LORE: An infrastructure to support location-aware services," *IBM Journal of Research and Development,* vol. 48, pp. 601-615, September 2004.

[76] S. Halliday. (10/5/2014). *Location aware software for mobile devices*. Available: https://github.com/fommil/openlapi

[77] S. J. Barbeau, M. A. Labrador, P. L. Winters, R. Pérez, and N. L. Georggi, "Location API 2.0 for J2ME – A new standard in location for Java-enabled mobile phones," *Computer Communications,* vol. 31, pp. 1091-1103, 2008.

[78] P. Campillo-Sanchez and J. Botia, "Simulation Based Software Development for Smart Phones," in *Ambient Intelligence - Software and Applications*, ed Springer Berlin: Springer Berlin / Heidelberg, 2012, pp. 243-250.

[79] M. Carbone and L. Rizzo, "Dummynet revisited," *ACM SIGCOMM Computer Communication Review,* vol. 40, pp. 12-20, 2010.

[80] *Android X86 Project*. Available: http://www.android-x86.org/ (accessed: December, 2015)

[81] V. Oracle, "VirtualBox," *User Manual–2013,* 2013.

[82] S. Canada. (2016). *New motor vehicle sales*. Available: http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/trade12-eng.htm

[83] S. Canada. (2016). *Statistics Canada, New automotive vehicles*. Available: http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/trad52-eng.htm

[84] C. Perkins, ""IP mobility support," Network Working Group, RFC 2002.

[85] H. Soliman, L. Bellier, K. Elmalki, and C. Castelluccia, "Hierarchical mobile IPv6 (HMIPv6) mobility management," 2008.

[86] R. Koodli, "Mobile IPv6 fast handovers rfc 5568," ed, 2009.

[87] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6 RFC 5213," 2008.

[88] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," in *IETF RFC 3963*, ed, 2005.

[89] S. Cespedes, S. Xuemin, and C. Lazo, "IP mobility management for vehicular communication networks: challenges and solutions," *IEEE Communications Magazine,* vol. 49, pp. 187-194, May 2011.

[90] R. Baldessari, W. i. Zhang, A. Festag, and L. Le, "A MANET-centric solution for the application of NEMO in VANET using geographic routing," in *Conference on Testbeds and research infrastructures for the development of networks & communities (TridentCom '08)*, Innsbruck, Austria, 2008, pp. 1-7.

[91] P. Bellavista, A. Corradi, and L. Foschini, "Context-aware handoff middleware for transparent service continuity in wireless networks," *Pervasive Mobile Comuputing,* vol. 3, pp. 439-466, August 2007.

[92] C. C. H. Soliman, K. El-Malki, and L. Bellier. (2005). *Hierarchical Mobile IPv6 Mobility Management (HMIPv6)*.

[93] H. Y. Jung, E. A. Kim, J. W. Yi, and H. H. Lee, "A scheme for supporting fast handover in hierarchical mobile IPv6 networks," vol. 27, pp. 798-801, 2005.

[94] C. Makaya and S. Pierre, "An Analytical Framework for Performance Evaluation of IPv6-Based Mobility Management Protocols," vol. 7, MARCH 2008.

[95] M. M. Torrent-Moreno, J. Santi, P. Hartenstein, H., "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information " *IEEE Transactions on Vehicular Technology,* vol. 58, pp. 3684 - 3703 September 2009.

[96] S. Bhaskar, G. Kumar, and P. Ramanareddy, "Signal strength based handover management protocol for future generation networks," in *Progress in Informatics and Computing (PIC), 2014 International Conference on*, 2014, pp. 556-561.

[97] J. Choi, Y. Khaled, M. Tsukada, and T. Ernst, "IPv6 support for VANET with geographical routing," in *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, 2008, pp. 222-227.

[98] W. Su, S. J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management,* vol. 11, pp. 3-30, 2001.

[99] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *Wireless Communications, IEEE,* vol. 13, pp. 36-43, 2006.

[100] K. Chodorow, *MongoDB: the definitive guide*: " O'Reilly Media, Inc.", 2013.

[101] M. Malensek, S. Pallickara, and S. Pallickara, "Evaluating Geospatial Geometry and Proximity Queries Using Distributed Hash Tables," *Computing in Science & Engineering,* vol. 16, pp. 53-61, 2014.

[102] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE,* vol. 7, pp. 12-18, 2008.

[103] *The Network Simulator - ns-2*. Available: http://www.isi.edu/nsnam/ns/ (accessed: December, 2015)

[104] H. Ko, G. Lee, D. Suh, S. Pack, and X. Shen, "An Optimized and Distributed Data Packet Forwarding Scheme in LTE/LTE-A Networks," 2015.

[105] M.-S. Pan, T.-M. Lin, and W.-T. Chen, "An enhanced handover scheme for mobile relays in LTE-A high-speed rail networks," *Vehicular Technology, IEEE Transactions on,* vol. 64, pp. 743-756, 2015.

[106] Q. Huang, J. Zhou, C. Tao, S. Yi, and M. Lei, "Mobile relay based fast handover scheme in high-speed mobile environment," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012, pp. 1-6.

[107] Y.-B. Lin, "Reducing location update cost in a PCS network," *IEEE/ACM Transactions on Networking (TON),* vol. 5, pp. 25-33, 1997.

[108] A.-C. Pang, Y.-B. Lin, H.-M. Tsai, and P. Agrawal, "Serving radio network controller relocation for UMTS all-IP network," *Selected Areas in Communications, IEEE Journal on,* vol. 22, pp. 617-629, 2004.

[109] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*: Elsevier Science, 2014.

[110] V. Jain and L. Venkatraman, "Method for performing proactive wireless communication handoffs using a mobile client's route information," ed: Google Patents, 2013.

[111] V. Jain, B. Raghunathan, and V. Kone, "Dead spot prediction method for wireless vehicular applications," ed: Google Patents, 2012.

[112] B. Raghunathan and V. Jain, "Dead spot mitigation methods for media applications in vehicular environments," ed: Google Patents, 2014.

[113] M. Akkouchi, "On the convolution of exponential distributions," *J. Chungcheong Math. Soc,* vol. 21, pp. 501-510, 2008.

[114] H. Jasiulewicz and W. Kordecki, "Convolutions of Erlang and of Pascal distributions with applications to reliability," *Demonstratio Mathematica. Warsaw Technical University Institute of Mathematics,* vol. 36, pp. 231-238, 2003.

APPENDICES

APPENDIX A – SUMMATION OF TWO RANDOM VARIABLES WITH MIXED ERLANG DISTRIBUTION

In this appendix, we derive the summation of two random variables with mixed Erlang distribution, the summation of several random variables with Gamma distribution can be evaluated as described in [113],[114]. Then the summation of Erlang distributions $S^{ERL}(t)$

$$S^{ERL}(\xi) = \sum_{j=1}^{n} \lambda_j^{m_j} e^{-\lambda_j t} \sum_{k=1}^{m_j} \frac{(-1)^{m_j-k}}{(k-1)!}\, \xi^{k-1}$$

$$\sum_{\substack{p_1+p_2+\cdots+p_n=m_j-k \\ p_j=0}} \prod_{\substack{l=1 \\ l\neq j}}^{n} \binom{m_l+p_l-1}{p_l} \frac{\lambda_l^{m_l}}{(\lambda_l-\lambda_j)^{m_l+p_l}} \quad (A.1)$$

and

$$S^{ERL}(\xi) = \mathcal{L}^{-1}\left\{\prod_{j=1}^{n}\left[\frac{\lambda_j}{\lambda_j+s}\right]^{m_j}\right\} \; for\ i\neq j\ \ and\ \lambda_i \neq\ \lambda_j\ \ (A.2)$$

However, we have two variables with mixed Erlang distribution, then the summation of two mixed Erlang variables can be evaluated as:

$$S^{MERL}(\xi)\ =\ \mathcal{L}^{-1}\left\{\left(\sum_{i=1}^{N_1}\alpha_{1,u}\left[\frac{\lambda_{1,i}}{\lambda_{1,i}+s}\right]^{m_{1,i}}\right)\left(\sum_{j=1}^{N_2}\alpha_{2,j}\left[\frac{\lambda_{2,j}}{\lambda_{2,j}+s}\right]^{m_{2,j}}\right)\right\}$$

$$(A.3)$$

$$S^{MERL}(\xi)\ =\ \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{2,j}\,\alpha_{1,i}\,\mathcal{L}^{-1}\left\{\left[\frac{\lambda_{1,i}}{\lambda_{1,i}+s}\right]^{m_{1,i}}\ \left[\frac{\lambda_{2,j}}{\lambda_{2,j}+s}\right]^{m_{2,j}}\right\}$$

$$(A.4)$$

For $\lambda_{1,i} = \lambda_{2,j}$ and $i = j$

$$S^{MERL}(\xi) = \sum_{i=1}^{N_1} \alpha_{2,i}\,\alpha_{1,i}\,\mathcal{L}^{-1}\left\{\left[\frac{\lambda_{1,i}}{\lambda_{1,i}+s}\right]^{m_{1,i}+m_{2,i}}\right\}$$

$$(A.5)$$

$$= \sum_{i=1}^{N_1} \alpha_{2,i}\,\alpha_{1,i}\,\frac{(\lambda_{1,i}\,t)^{m_{1,i}+m_{2,i}-1}}{(m_{1,i}+m_{2,i}-1)!}\,\lambda_{1,i}\,e^{-\lambda_{1,i}\,\xi}\ (A.6)$$

By comparing$(A.2)$ and $(A.4)$ and using$(A.1)$ the summation of mixed Erlang variables with different rates can be defined as

$$S^{MERL}(\xi) =$$

$$\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \alpha_{2,j}\,\alpha_{1,i}\,\left[g_{1,i}(\xi) + g_{2,j}(\xi)\right] \quad i \neq j\ and\ \lambda_i \neq \lambda_j\ \ (A.7)$$

$$g_{j,u}{}^{ERL}(\xi) = \lambda_{j,u}^{m_{j,u}}\,e^{-\lambda_{j,u}\xi} \sum_{k=1}^{m_{j,u}} \frac{(-1)^{m_{j,u}-k}}{(k-1)!}\,\xi^{k-1}\,\varphi_{j,k,u} \quad (A.8)$$

$$\varphi_{j,k,u} = \sum_{\substack{p_{1,u}+p_{2,u}=m_{j,u}-k \\ p_{j,u}=0}} \prod_{\substack{l=1 \\ l\neq j}}^{2} \binom{m_{l,u}+p_{l,u}-1}{p_{l,u}} \frac{\lambda_{l,u}^{m_{l,u}}}{(\lambda_{l,u}-\lambda_{j,u})^{m_{l,u}+p_{l,u}}}\ \ (A.9)$$

As shown the summation of two mixed Erlang random variables given by $(A.6)$ and $(A.7)$ results in an exponential function of ξ multiplied by ξ^{k-1} and a scalar quantity $\varphi_{j,k,u}$. Using these equations exact distribution of the sum of two mixed Erlang variables can be found.

APPENDIX B – SIMPLIFICATION OF THE EXPECTED NUMBER OF OVERHEAD PACKETS

In this appendix, we simplify $E[N_{OP}]$ to be in closed form and suitable for numerical evaluation.

$$E[N_{OP}] = \sum_{i=1}^{\infty} 1 - \sum_{n=0}^{i-1} \frac{\lambda_{arrival}^{n}}{n!} \left[(-1)^n \frac{d^n}{ds^n} f_\theta^*(S)\right]_{S=\lambda_{arrival}} \quad (\mathrm{B.1})$$

Let

$$g_\theta^*(n) = \frac{\lambda_{arrival}^{n}}{n!} \left[(-1)^n \frac{d^n}{ds^n} f_\theta^*(S)\right]_{S=\lambda_{arrival}} \quad (\mathrm{B.2})$$

From Taylor's series

$$f_\theta^*(0) = \sum_{n=0}^{\infty} g_\theta^*(n) \qquad (\mathrm{B.3})$$

$$E[N_{OP}] = \sum_{i=1}^{\infty} 1 - \left[\sum_{n=0}^{\infty} g_\theta^*(n) - \sum_{n=i-1}^{\infty} g_\theta^*(n)\right] \quad (\mathrm{B.4})$$

$$= \sum_{i=1}^{\infty} \left[1 - \sum_{n=0}^{\infty} g_\theta^*(n) + \sum_{n=i-1}^{\infty} g_\theta^*(n)\right]$$

$$= \sum_{i=1}^{\infty} 1 - f_\theta^*(S)|_{S=0} + \sum_{n=i-1}^{\infty} g_\theta^*(n) \quad (\mathrm{B.5})$$

$f_\theta^*(S)|_{S=0} = 1$ *for mixed Erlang then*

$$E[N_{OP}] = \sum_{i=1}^{\infty} \sum_{n=i-1}^{\infty} g_\theta^*(n) \qquad (\mathrm{B.6})$$

$for\ i = 1$

$$E[N_{OP}] = \sum_{n=0}^{\infty} g_\theta^*(n) = f_\theta^*(0) \qquad (\mathrm{B.7.1})$$

$for\ i = 2$

$$E[N_{OP}] = \sum_{n=1}^{\infty} g_{\theta}^{*}(n) = f_{\theta}^{*}(0) - g_{\theta}^{*}(0) \quad (\mathrm{B.7.2})$$

$for\ i = 3$

$$E[N_{OP}] = \sum_{n=2}^{\infty} g_{\theta}^{*}(n) = f_{\theta}^{*}(0) - g_{\theta}^{*}(0) - g_{\theta}^{*}(1) \quad (\mathrm{B.7.3})$$

Then,

$$E[N_{OP}] = \lim_{n\to\infty} n\big(f_{\theta}^{*}(0)\big) - (n-1)g_{\theta}^{*}(0) - (n-2)g_{\theta}^{*}(1) - (n-3)g_{\theta}^{*}(2) - \cdots \qquad (\mathrm{B.8})$$

$$E[N_{OP}] = \lim_{n\to\infty} n\Big(f_{\theta}^{*}(0) - \big(g_{\theta}^{*}(0) + g_{\theta}^{*}(1) + g_{\theta}^{*}(2) + \cdots\big)\Big) + \big(g_{\theta}^{*}(0) + 2g_{\theta}^{*}(1) + 3g_{\theta}^{*}(2) + \cdots\big) \qquad (\mathrm{B.9})$$

$$E[N_{OP}] = \big(g_{\theta}^{*}(0) + 2g_{\theta}^{*}(1) + 3g_{\theta}^{*}(2) + \cdots\big) \quad (\mathrm{B.10})$$

$$= \left(f_{\theta}^{*}(S) + 2\,\lambda_{arrival}\frac{d}{ds}\,f_{\theta}^{*}(S) + 3\,\frac{{\lambda_{arrival}}^{2}}{2!}\frac{d^{2}}{ds^{2}}\,f_{\theta}^{*}(S) + \cdots\right) \qquad (\mathrm{B.11})$$

$$However, \qquad \left.\frac{d}{ds}\,f_{\theta}^{*}(S)\right|_{s=\lambda_{arrival}} = \sum_{n=0}^{\infty}\frac{\lambda_{arrival}^{n}}{n!}\left[(-1)^{n}\frac{d^{n+1}}{ds^{n+1}}\,f_{\theta}^{*}(S)\right]_{S=\lambda_{arrival}} \quad (\mathrm{B.12})$$

$Then$

$$E[N_{OP}] = \left.f_{\theta}^{*}(S)\right|_{s=0} - \lambda_{arrival}\left.\frac{d}{ds}\,f_{\theta}^{*}(S)\right|_{s=\lambda_{arrival}} \quad (\mathrm{B.13})$$