Towards High Quality Video Streaming over Urban Vehicular Networks Using A Location-Aware Multipath Scheme

Renfei Wang

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the M.A.Sc. degree in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

The transmitting of video content over Vehicular Ad Hoc Networks (VANETs) faces a great number of challenges caused by strict QoS (Quality of Service) requirements and highly dynamic network topology. In order to tackle these challenges, multipath forwarding schemes can be regarded as potential solutions. However, route coupling effect and the path length growth severely impair the performance of multipath schemes.

In this thesis, the current research status about video streaming over VANETs as well as multipath transmissions are reviewed. With the demand to discover a more suitable solution, we propose the Location-Aware Multipath Video Streaming (LIAITHON$^+$) protocol to address video streaming over urban VANETs. LIAITHON$^+$ uses location information to discover relatively short paths with minimal route coupling effect. The performance results have shown it outperforms the underlying single path solution as well as the node-disjoint multipath solution. In addition, the impact of added redundancy on the multipath solution is investigated through LIAITHON$^+$. According to the results, added redundancy has a different impact depending on the data rate.

Acknowledgements

First and foremost, I am deeply indebted to my supervisor Prof. Azzedine Boukerche, who has provided me with the consistent guidance and extraordinary patience, as well as financial support, throughout my entire master's study. It is no overstatement to say that my experience as a student with such a charismatic supervisor will be greatly treasured for my whole life.

My special thanks goes to Cristiano Rezende, who has alway been there whenever I faced any questions in my research. His valuable advice, constructive criticism and enduring encouragement made this work possible. I sincerely wish him all the best in his future academic career.

I would like to extend my heartfelt gratitude to all my colleagues at PARADISE Research Laboratory. Everyone's selfless and generous care makes the whole lab full of sunshine.

Last but not the least, I am particularly grateful to my parents and my friends for their endless help and support.

Publications Related to Thesis

Conference Paper:

- Renfei Wang, Cristiano Rezende, Heitor S. Ramos, Richard W. Pazzi, Azzedine Boukerche, Antonio A.F. Loureiro: "LIAITHON: A location-aware multipath video streaming scheme for urban vehicular networks". Accepted in the Seventeenth IEEE Symposium on Computers and Communication (ISCC'12), to be held in July 1 - 4, 2012, Cappadocia, Turkey.

Journal Paper:

- Mohammed Almulla, Renfei Wang, Cristiano Rezende, Heitor S. Ramos, Richard W. Pazzi, Azzedine Boukerche, Antonio A.F. Loureiro: "Towards High Quality Video Streaming over Urban Vehicular Networks Using A Location-Aware Multi-path Scheme". Submitted to the IEEE Transactions on Computers (TC), 2012.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As an emerging research area, vehicular networks have received a lot of attention in recent years. The communication in vehicular networks consists of vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) [1, 2]. In order to achieve a V2I network, communication infrastructures, such as RSUs (Road Side Units), have to be built along the road at every short road segment. This incurs significantly high deployment and maintenance costs. However, even with excessive investments, there still exists many in-between vehicles which are not covered by any infrastructures. In this case, V2V networks, which do not rely on any road side infrastructure, are preferred. V2V communication is purely based on the ad hoc network between vehicles; thus, it is also known as VANET (Vehicular Ad Hoc Network).

In support of the rapid development of vehicular networks, standards, mainly developed in DSRC (Dedicated Short-Range Communications) standard suite [3] have been developed to accommodate for VANETs' requirements. As a core standard in DSRC, IEEE 802.11p [4] focuses on the physical (PHY) and medium access control (MAC) layer for the purpose of including vehicular environments in the 802.11 standard. Heavily based on IEEE 802.11p, WAVE (Wireless Access in Vehicular Environments) [5] is developed, which contains IEEE 1609 family (1609.1[6], 1609.2[7], 1609.3[8], 1609.4[9]).

These standards directly aim to provide a fundamental base for applications which have been proposed to enable the idea of an Intelligent Transportation System (ITS). The potential applications in vehicular networks can be classified into the following categories [10]:

- Road safety applications, which aim to reduce the number of accidents as well as to improve the emergency response in eventual accidents. Sample applications can be intersection movement assistance, lane change warning and emergency electronic braking.
- Traffic efficiency applications, which mainly aim to control the traffic congestion as well as to reduce the fuel consumption. Sample applications can be route guidance and green light optimal speed advisory.
- Infotainment applications, which are proposed to provide information and entertainment services to the drivers and passengers. Sample applications can be roadside advertising, podcasting and online gaming.

1.1 Motivation

Obviously, enabling video streaming over vehicles provides basic support for infotainment such as gaming and advertising. As a mater of fact, the other categories will also be greatly enhanced by the video support. Imagine there is an accident, which causes a heavy traffic jam in a downtown area. For enhancing traffic safety, timely video information about the accident scene can be transmitted to incoming first responders; this enables them to prepare before they reach the site. For the sake of transport efficiency, the following drivers are able to receive video information concerning what is happening in front of them and so they may judge whether it is necessary to take a detour accordingly. Meanwhile, by using infotainment applications, the passengers can enjoy online video content or gaming while waiting on the road.

Apart from normal video transmission, video streaming requires the video content to be presented while still being transmitted from the video provider [11]. Due to the special characteristics of video streaming, it can be used to provide timely video information without the support of large storage (only limited storage for buffering is needed). As a mater of fact, the delay requirement of video streaming is much more stringent when compared to other data transmission. Moreover, due to that, retransmission is almost impossible. This requires the video streaming protocol to provide a high delivery ratio within a limited time. This task also faces harsh challenges due to the dynamic nature of VANETs. Therefore, the multipath transmission technique is introduced in the proposed method in order to better tackle this problem.

1.2 Problem Statement

The VANETs' environment as well as the stringent QoS (Quality of Service) requirements makes video streaming over VANETs challenging. Single path works addressing this topic are prone to collision when the data rate goes higher. By using multipath transmissions, this problem can be solved by distributing the heavy traffic load of video content among a set of paths. However, most of the existing multipath works cannot be adapted to the highly dynamic topology of VANETs. Moreover, few works take into consideration the route coupling effect and path length growth. Thus a suitable multipath solution considering these issues is required. Moreover, in search of further improvements, research on the impact of added redundancy is conducted.

1.3 Thesis Objective

The main objective of this thesis is to discover a suitable solution for the task of video streaming over VANETs. Towards this objective, the concept of using multiple paths is incorporated into the receiver based forwarding scheme. The forwarding nodes are chosen based on its waiting time which depends on greedy and link durability, as well as the multipath metric. For the purpose of investigating the appropriate path number, two approaches are designed and implemented. Furthermore, two methods are proposed to investigate the impact of added redundancy on the proposed multipath solution.

1.4 Contributions

The major contributions of this thesis are:

1. LIAITHON$^+$ (Location-Aware Multipath Video Streaming), a brand new multipath transmission model for video streaming, is proposed. It neatly fills the gap of the receiver based multipath forwarding scheme. The simulations have proven that it is very adaptable to video content delivery over high mobility VANETs.

2. Two brand new solutions, 2-path LIAITHON$^+$ and 3-path LIAITHON$^+$, are designed and implemented based on the model of LIAITHON$^+$; each of these solutions uses a innovative area-aware waiting time calculation scheme and route coupling prevention scheme. Based on these schemes, the impact of path number on multipath video streaming is investigated through the simulations.

3. The impact of added redundancy on the multipath scheme is investigated through two proposed solutions, 2LR and 3LCR.

1.5 Thesis Organization

In this chapter, the background information about video streaming over VANETs is introduced. After this, the problem statement is discussed, followed by the thesis objective and main contributions. The remainder of this thesis is structured as follows:

- Chapter 2 presents existing works relevant to the proposed method. To better illustrate the related works, they are classified into two parts: video streaming over

VANETs and multipath transmission over ad hoc networks.

- Chapter 3 reviews the underlying single path solution, VIRTUS. The important characteristics along with the core design concepts are introduced.

- Chapter 4 fully describes the proposed method, LIAITHON$^+$. First, the motivation of the proposed method is provided. After that, the streaming process as well as the core metric, degree of closeness, are described in detail. Relying on these concepts, the design issues of 2-path and 3-path LIAITHON$^+$ are elaborated. Finally, the simulation results are explained and discussed.

- Chapter 5 proposes two redundancy solutions based on LIAITHON$^+$ for the purpose of investigating the impact of added redundancy on the multipath solution.

- Chapter 6 provides the conclusion for this thesis, as well as some suggestions for future work.

Chapter 2

Related Work

This chapter summarizes most of the existing works related to the proposed solution, LIAITHON$^+$. Some of them such as performance or impact studies provide background information or major challenges for LIAITHON$^+$; other studies, however, tackle similar problems as LIAITHON$^+$. Since there are few works directly addressing multipath video streaming over VANETs, this chapter is divided into two parts: video streaming in VANETs and multipath transmission over ad hoc networks. For each part, the motivation as well as challenges are first described, then the related protocols are briefly illustrated.

2.1 Video Streaming over VANETs

As introduced in Chapter 1, video streaming provides basic support for a wide variety of potential applications in VANETs, including road safety, traffic efficiency and infotainment. Fortunately, it is feasible to stream video over vehicles. Abundant electricity supplied by vehicle engines can provide ample energy for video transmission and playback. Also, enough room is available inside vehicles to accommodate large on-board computational devices and storage for video encoding and decoding.

However, by using vehicles as the intermediate forwarding nodes to transmit video content, a lot of tricky challenges arise from the VANETs' environments. Communication between any source-destination pair in an ad hoc network is composed of multiple hops. Many studies have shown that the end-to-end delay is almost proportional to the number of hops [12]. This severely lowers the adaptability of video streaming over long distance in VANETs, especially in the case of safety-related applications [13]. Another QoS problem with multihop is the packet loss probability increases as the hop number increases [12]. A simple explanation is given as follows. Assume each link has a 0.1 probability of packet loss. The total loss probability for a 5 hop transmission is $1 - (1 - 0.1)^5$, which approximately equals 0.41. Therefore, to fulfill the QoS requirements, streaming solutions should take hop numbers into consideration (e.g., [14]).

Apart from the traditional challenges of ad hoc networks, highly dynamic topology is regarded as the major problem of VANETs [13, 15]. This problem should be handled by all the data transmission protocols, especially protocols related to video transmission . The speed of vehicles can reach 50km/h in urban cities while it may exceed 100km/h in highway conditions. This means the relative speed between two running vehicles will be significantly high. Thus, the connections between vehicles will be more transient when compared to other ad hoc networks such as MANETs (mobile ad hoc networks). For this reason, solutions for VANETs need to be concerned with the movements of vehicles (e.g., [16]).

Moreover, the non-uniform distribution of vehicles incurs a great amount of problems for schemes proposed in VANETs [17]. For instance, the density of vehicles in the urban areas will be largely increased during rush hour, which leads to network collisions if most of the vehicles share the same communication channel. However, the density in suburban areas is extremely low, which results in severe network partitions. Since it is easy to find decoupled multiple paths in the urban scenario, the network collision problem is our major focus.

Furthermore, challenges will be encountered when using VANETs' applications in real scenarios. Without roadside infrastructures, vehicle to vehicle networks largely depend on the penetration ratio, which refers to the ratio of vehicles equipped with communication

Table 2.1: QoS requirements of video transmission

QoS metrics	Video Streaming	Interactive Video
Delivery Ratio	$< 5\%$	$< 1\%$
Delay	$4 \sim 5$ seconds	< 150 ms
Jitter	No significant requirements	< 30 ms

capability to total vehicles [18]. Clearly, a low penetration ratio will lead to network disconnections. However, even with a high penetration ratio, there still exists many black holes such as parks or lakes, inside which no communication unit (i.e., vehicles) is located. Moreover, obstacles such as skyscrapers cannot be penetrated by radio waves, which causes network disconnections as well. Therefore, only solutions which are highly reactive to the network topology, such as receiver based solutions which are described in Subsection 2.1.2, are preferred.

2.1.1 QoS Requirements of Video Streaming

These challenges from VANETs are exacerbated when they face the stringent QoS requirements of video streaming. The network QoS requirements of video transmission is studied in [19]. Unicast video transmission is categorized into video streaming and interactive video. Interactive video is defined as bidirectional streams between two nodes. For example, video conference can be regarded as the typical application of interactive video. The QoS requirements for both types are shown in table 2.1. As we can observe, although less stringent than the requirements of interactive video, the requirements of video streaming is extremely high in terms of delivery ratio and delay. This makes it difficult to achieve video streaming over vehicular ad hoc networks.

2.1.2 Related Protocols

Among numerous works attempting to stream data over ad hoc networks, only a few works are able to adapt to the highly dynamic environment of VANETs. Geographic routing protocol, through its use of the physical location of nodes[1], is considered the most successful approach in VANETs' scenarios [20, 21]. Authors in [22] have categorized the unicast geographic routing protocols, based on the forwarding strategy, into greedy, opportunistic and trajectory based strategies. Greedy strategy is used to choose the next forwarder (i.e the node responsible for forwarding packet) which is as close to the destination as possible, to shorten the end-to-end delay. This is quite applicable for video streaming in terms of the delay of QoS. On the contrary, opportunistic strategy is not suitable for video streaming, since it is based on the store-carry-forward scheme, which can only be used for delay tolerant applications. Trajectory based strategy uses predefined trajectory to transmit the packets. Like source routing, the source in trajectory based strategy discovers the optimal route and appends the route information onto the packet's header. The route information only refers to geometrical lines instead of specified forwarders as in source routing.

In terms of forwarding schemes, authors in [23] classify different data forwarding schemes for VANETs into traditional route based forwarding (e.g., AODV (Ad-hoc On-Demand Distance Vector Routing) [24] and DSR (Dynamic Source Routing) [25]), sender based forwarding (SBF) and receiver based forwarding (RBF). The traditional route based forwarding schemes originated from MANETs. Although VANETs share many similarities with MANETs, many studies demonstrate that traditional route based forwarding schemes are not efficient in high mobility scenarios. SBF refers to the scheme in which forwarding nodes are chosen by senders. On the contrary, the forwarders in RBF are chosen by the receivers. In both traditional route based forwarding and SBF, periodical beacon messages, such as location and movement information, are exchanged between neighboring nodes, in order for each node to be aware of its neighbors. However, this beaconing method incurs high overhead. Furthermore, [23] evaluates the performance of RBF and SBF based on simulations. It is shown that RBF outperforms SBF within VANETs' environments. The main reason for this is RBF does not rely on any neigh-

[1]Nodes refer to the vehicles in this thesis.

Table 2.2: Data forwarding zone of V3

Current Location	Driving Direction	Current Time	In the DFZone
$*$	towards r	before boundary	Yes
$*$	$*$	after boundary	No
$dist(r, v) < dist(r, s)$	away from r	before boundary	Yes
$dist(r, v) > dist(r, s)$	away from r	$*$	No

boring information that may easily become outdated in a high mobility scenario.

In the following subsections, some typical works related to our proposed method will be introduced briefly.

2.1.2.1 V3

A well-known geographic protocol, which uses RBF for video streaming, is proposed in V3 (Vehicle-to-Vehicle live Video streaming) [26]. The author proposes a forwarder selection method by means of a forwarding zone. Table 2.2 shows the criteria used to determine the data forwarding zone (DFZone).$*$ means this entry can be any value; r, s and v refer to video requester, video source and the intermediate node respectively; $dist()$ is a operation which calculates the distance between its two operands. And, the time boundary is calculated by using the deadline minus the estimated arrival time at the video requester. Upon receiving a video packet, the intermediate node v will calculate whether it is located in the data forwarding zone (DFZone). If so, it will forward the package immediately. Otherwise, it will buffer the packet and make a forwarding zone recalculation after a certain time interval (store-carry-forward). The major problem with this scheme is that multiple forwarders can be generated based on the criteria, which leads to redundancy and collisions. On the other hand, if no receiver satisfies the criteria, the video content is buffered which causes unacceptable delays.

2.1.2.2 Fast Triggering

A video triggering solution which uses RBF scheme as well as greedy strategy is proposed in fast triggering [14]. The triggering of video delivery as quickly as possible is an important issue for safety-related video streaming. In order to shorten the delay, the scheme makes use of the greedy strategy, in which the chosen forwarder is always close to the transmission range border of its last hop (previous forwarder). In order to precisely estimate the transmission range, fast triggering uses periodical hello message to estimate the actual frontward and backward transmission range. Once a node receives a packet, it will schedule itself to forward the received packet based on its waiting time. If a node overhears transmission for the same packet before its scheduled time, it will cancel the forwarding schedule. The waiting time of a certain node n_i is randomly chosen within the time contention window (TCW), which is calculated as follows:

$$TCW = \left[\left(\frac{ETR - Dist}{ETR} \times (TCWMax - TCWMin)\right) + TCWMin\right] \tag{2.1}$$

where ETR is estimated transmission range and $Dist$ is the distance between n_i and its last hop. $TCWMax$ and $TCWMin$ are two parameters which refer to the maximum and minimal value of the time contention window, respectively. As we can observe from the above equation, the closer the node is to the border of the transmission range, the greater the possibility it may become the forwarder.

A similar greedy approach is proposed in [27]. Although these approaches aim to reduce the end-to-end delay, they may not achieve low delay since every single forwarding introduces waiting time. This could be reduced if a chosen forwarder is used to transmit more than one packet. However, the link between forwarders is too vulnerable to withstand continuous packets' transmission, because the chosen forwarders are often close to the border of the transmission range. Thus, only using greedy strategy to calculate waiting time is not enough for video streaming.

2.1.2.3 MOPR-GPSR

GPSR (Greedy Perimeter Stateless Routing) [21], a well-known greedy geographic protocol, uses the location information of the intermediate node, its neighbors and the destination to discover the closest forwarder in terms of the distance towards destination. When such a node is not available, the packet will be routed around the void area by perimeter mode.

Based on GPSR, authors in [28] proposed a routing scheme, namely MOPR-GPSR, by using the concept of MOPR (Movement Prediction-based Routing). MOPR is first presented in [16], in which the underlying protocol is AODV. MOPR uses the velocity and the location information of two nodes to calculate the lifetime of their link, which measures the link durability. The lifetime (δ) of the link between node i and node j can be calculated by solving the following equation:

$$TR^2 = ((X_i + Vx_i \times \delta) - (X_j + Vx_j \times \delta))^2 + ((Y_i + Vy_i \times \delta) - (Y_j + Vy_j \times \delta))^2 \quad (2.2)$$

where TR is the transmission range. The location vector components of node i and node j are (X_i, Y_i) and (X_j, Y_j) respectively, and the velocity vector components are (Vx_i, Vy_i) and (Vx_j, Vy_j) respectively. This work takes link durability into consideration, which is very suitable for long term data transmission such as video streaming. However, the lifetime calculation is based on the constant velocity model [29] which is very unrealistic.

2.1.2.4 SiFT

By using RBF, a trajectory based solution is proposed in SiFT (Simple Forwarding over Trajectory) [30]. Upon receiving a packet, a node schedules itself to forward the packet based on the waiting time calculated. If no other transmission, for the same packet, is detected during the waiting time, this node will forward the packet. The waiting time (WT) is calculated as follows:

$$WT = \alpha \times \frac{D_t}{D_l} \tag{2.3}$$

where α is a constant, D_t is the distance from the node to the trajectory and D_l is the distance from the node to its last hop. According to the equation, the optimal forwarder should be close to the trajectory while distant from the last hop. The authors further exploit a scheme to prevent duplicated packets forwarding, by limiting D_t within a threshold called admissibility strip. SiFT uses both metrics for greedy (D_l) and for trajectory (D_t) to select the forwarder; however, the waiting time equation for balancing the two metrics is not reasonable enough. For instance, node A ($D_l = 0.01$, $D_t = 1$) and node B ($D_l = 0.1$, $D_t = 10$) have the same waiting time; but, node A is actually much more suitable for the task of forwarding packets.

2.1.3 Summary

Video streaming is mainly motivated by a variety of potential applications tailored for VANETs. However, enabling video streaming over vehicles faces many challenges from VANETs' environments, such as the highly dynamic topology. Furthermore, in order to fulfill the stringent QoS requirements of video streaming, these challenges are stressed to the maximum. Under this circumstance, greedy and trajectory based geographic routing protocols are preferred. In terms of the forwarding scheme, receiver based forwarding is suggested.

Table 2.3 compares the protocols described above with our underlying protocol, VIRTUS. VIRTUS is a receiver based streaming scheme that relies on the greedy strategy. Moreover, it takes link durability into consideration in order to solve the unstable link problem of greedy strategy. These characteristics make VIRTUS very suitable for the task of video streaming over VANETs; thus, it is chosen as the underlying scheme for our proposed solution LIAITHON$^+$. VIRTUS will be described in Chapter 3 in detail. Our proposed method, LIAITHON$^+$, does not only inherit all the characteristics of VIRTUS,

Table 2.3: Comparison of VANETs' protocols

Protocol	Forwarding Strategy	Forwarding Scheme
V3 [26]	Greedy & Opportunistic	Receiver based
Fast Triggering [14]	Greedy	Receiver based
GPSR [21]	Greedy	Sender based
MOPR-GPSR [28]	Greedy & Link Durability	Sender based
SiFT [30]	Trajectory based	Receiver based
VIRTUS [15]	Greedy & Link Durability	Receiver based

but it also uses the trajectory based strategy to discover decoupled multiple paths.

2.2 Multipath Transmission over Ad Hoc Networks

Multipath transmission is regarded as one of the most promising techniques which can be used to improve the performance of video transmission over ad hoc networks. By using multiple paths to deliver video content simultaneously, performance can be increased from several aspects. Congestions and collisions, especially under high data rate, can be controlled and reduced by distributing data load among several paths. Path diversity will provide fault tolerance while one path is disconnected. Moreover, recovery packet can be transmitted onto the additional path in order to cover the packet loss. Some works [31, 32] also suggest the use of multistream coding coupling with multipath transmission in order to protect important frames by transmitting them over the most reliable paths discovered.

Although the benefits of multipath scheme are quite impressive, delivering video content along multiple paths introduces a wide variety of difficulties as well [33]. Some problems shown in studies, such as route request storm [34], are only incurred from

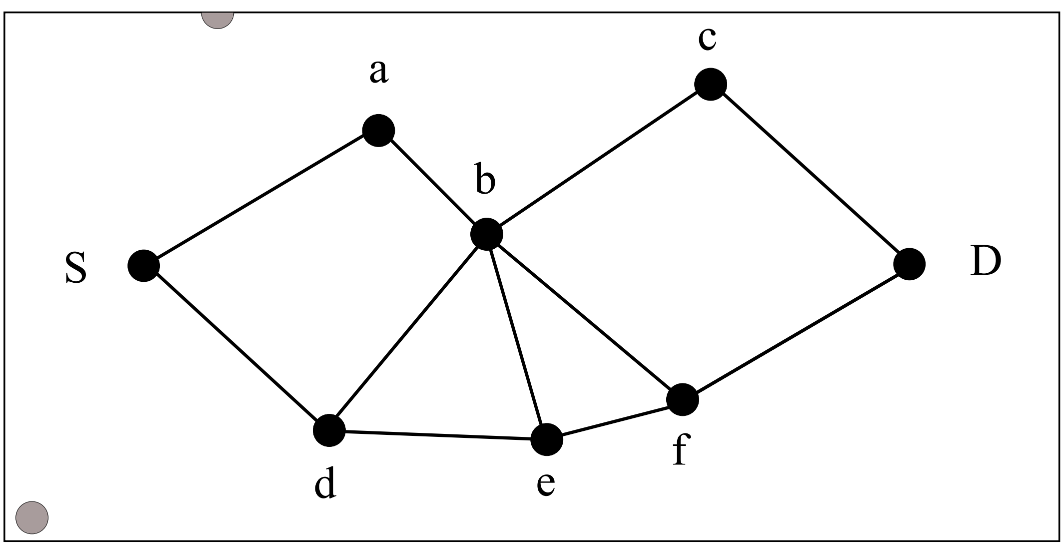


Figure 2.1: Path diversity

specific type of protocols. Others are inherent issues related directly to the multipath method, regardless of what protocols are used. They include, but are not limited to, the way to discover diverse paths, the route coupling effect and the increase of the path length.

The chosen paths should be as diverse as possible in order to prevent them from being affected by the same network disruption. In other words, a set of diverse paths always provides more fault tolerance. Most of the existing multipath works suggest the use of disjoint paths to achieve path diversity. The path disjointness can be classified into three types: they are non-disjoint, link-disjoint and node-disjoint. For a source-destination pair, the link-disjoint refers to the set of chosen paths that do not share common links. As shown in Figure 2.1, path S-a-b-c-D and path S-d-b-f-D are two link-disjoint paths. The link-disjoint paths are prone to collision at the common node. For example, node a and node b may simultaneously transmit packet to node b which causes packet loss. A more diverse pair of paths are S-a-b-c-D and S-d-e-f-D. There is no common nodes shared by the two paths, thus they are known as node-disjoint paths. Compared to link-disjoint, node-disjoint paths provide more fault tolerant. Moreover, authors in [35] analyzed the impact of path disjointness on route stability and concluded that node-disjoint paths are more stable than link-disjoint paths. Therefore, node-disjoint paths are preferred in many works [35–38].

The major problem of using multipath is the route coupling effect, since most works for ad hoc networks are based on the single channel model. The forwarders on the neighboring paths will contend for access to the shared channel while they are in each other's communication range. Such collisions between multiple paths are defined as the route coupling effect. The widely used node-disjoint paths cannot solve this problem at all. As can be observed in Figure 2.1, the node b on path S-a-b-c-D and the nodes d, e and f on path S-d-e-f-D share the same channel which may lead to severe collisions between the two paths.

Several performance studies [39–42] on multipath solutions are provided, which estimate the impact of the route coupling effect. [39] explains and simulates the impact of route coupling effect on alternate path routing (APR). Although APR is able to alleviate the traffic load by distributing packets among a set of paths, the simulation shows the result of using APR only has a negligible affect if the paths are not isolated enough. [40] simulates the node-disjoint multipath solutions for VANETs with different interference levels. The performance evaluation has shown that the benefit of using node-disjoint paths is only revealed when the chosen paths have a low interference level. [42] evaluates the bandwidth of four different interference models, while [41] estimates and compares the throughput of single path and two path solutions under single and multiple source-destination pair scenarios. Both [42] and [41] have concluded that multipath transmission is impractical if the route coupling is not controlled. In other words, all of these performance studies have indicated the noticeably negative impact of the route coupling effect.

The demand of path diversity and route coupling prevention leads to a significant increase in path length compared to the greedy single path solution. The increase of hop numbers may cause unacceptable delays and increased packet loss probability for video streaming. Thus, there exists a trade-off between keeping all paths spatially disjointed from each other and keeping them close to the shortest path.

2.2.1 Related Protocols

Multipath solutions addressing data transmissions in ad hoc networks are mainly studied in MANETs. Most of the existing works (e.g., [35, 36, 38, 43–45]) are based on the traditional route based forwarding scheme while they employ a modified route discovery and route maintenance method. Only a few works (e.g., [37, 46]) have addressed the multipath issue specifically for VANETs. Moreover, few approaches (e.g., [47, 48]) have considered the route coupling effect. In the following subsections, some typical works are briefly explained.

2.2.1.1 SMR

SMR (Split Multipath Routing) [36], a well-known multipath work based on DSR [25], is focused on discovering a pair paths which are maximally node-disjoint in ad hoc networks. It uses source routing to forward the packets. When a source node needs to send a packet, it first sends a RREQ (route request) message by flooding it to the whole network. Every intermediate forwarder will append its node ID onto the packet's header in order to record path information. Since the packet is flooded, the destination will get several duplicated RREQs. Each RREQ contains the path information indicating the set of forwarders' ID along the way. The destination then selects maximally node-disjoint paths, on which the RREP (route reply) messages are transmitted back to the source. Upon receiving the multiple RREPs, the source uses the paths information inside RREPs to transmit the packet.

In order to increase the possibility of having diverse paths discovered in the RREQ session, unlike the DSR, the intermediate nodes which know the route to the destination are not allowed to send RREPs to the source. Moreover, they will forward the duplicate RREQs which are from different last hops, as long as their hop number are less than the first RREQ received. However, this will lead to a route request storm which causes higher overhead and collision rate. The solution limits its path number to two. One is the path taken by the first arrival RREQ. The other path is the one which is disjointed maximally from the first path. The protocol also uses a simple route maintenance scheme to adapt the mobility scenario; however, this protocol is still very inefficient for high mobility VANETs.

2.2.1.2 AOMDV & HM-AOMDV

As a multipath extension to AODV, AOMDV (Ad hoc On-demand Multipath Distance Vector) is proposed in [43]. Unlike the source routing used in SMR, AOMDV uses the hop-by-hop routing approach, in which every node uses its route table to forward packet. In order to support multipath, every node keeps a route_list in its routing table instead of next_hop such as in AODV. The route_list is comprised of a list of next hops along with their hop numbers to the destination. Like AODV, each entry of the route_list is recorded during the route advertising (RREQ and RREP) session. Entries in the route_list have the same destination sequence number, and will be removed when the node receives advertisement with a higher destination sequence number. In order to ensure that each intermediate node's route_list represents a set of node-disjoint paths, the following method is used. Instead of immediately discarding duplicated copies of RREQ such as in AODV, the intermediate node will check if they are sent via different neighbors of the source from that of the recorded RREQ. If so, it will update its route_list by inserting the new last hop as the next hop on the reverse path. Moreover, every intermediate node only forwards the first RREQ it receives.

AOMDV attempts to discover link-disjoint paths, since the number of node-disjoint paths are considered to be limited. Thus, a "looser" reply policy is used in the destination node. To be specific, the destination transmits several duplicated RREP messages, regardless of whether they are from different neighbors of the source, to its different neighbors respectively. After that, the RREPs are relayed on the reverse paths set by

RREQs. Along with their transmissions, the route_lists to the destination are built up.

Based on AOMDV, HM-AOMDV [37] proposed a multipath solution for VANETs by taking into consideration the vehicle mobility factor. Such factor is measured according to the relative velocity between vehicles. However, it is still built upon a traditional route based scheme, which is considered to have low performance in VANETs.

2.2.1.3 SDMR

Authors in SDMR (Spatially Disjoint Multipath Routing)[48] suggest the use of spatially disjoint paths to maximally avoid route coupling effect. Similar to SMR, the intermediate node will append its ID onto the RREQ in order to record the route. Also, duplicated RREQs are allowed to be forwarded by intermediate nodes. After the destination receives the RREQs, it replies with RREPs to the source via reverse paths. The intermediate nodes will add the locations of their 1-hop neighbors, perceived through HELLO messages, into the RREPs. In this case, the source node is able to build up a partial network graph. Based on this graph, it chooses the most spatially disjointed paths from the path distance metric.

The distance between path p_i and path p_j is estimated as the weighted average distance from nodes (excluding source and destination) on p_i to p_j. The distance from a node n_k to the p_j is represented as d_k; and, it is chosen as the minimal distance from n_k to p_j in terms of hop number. The calculation is shown as follows:

$$distance(p_i, p_j) = \sum_{k=1}^{l} d_k \times w_k \; ; \; \sum_{k=1}^{l} w_k = 1 \tag{2.4}$$

where w_k is the weighting factor and l refers to the number of hops on path p_i (excluding source and destination).

Although this solution successfully avoided the route coupling between paths, the partial network graph is notably inaccurate under the high mobility scenario which limits its usage for VANETs.

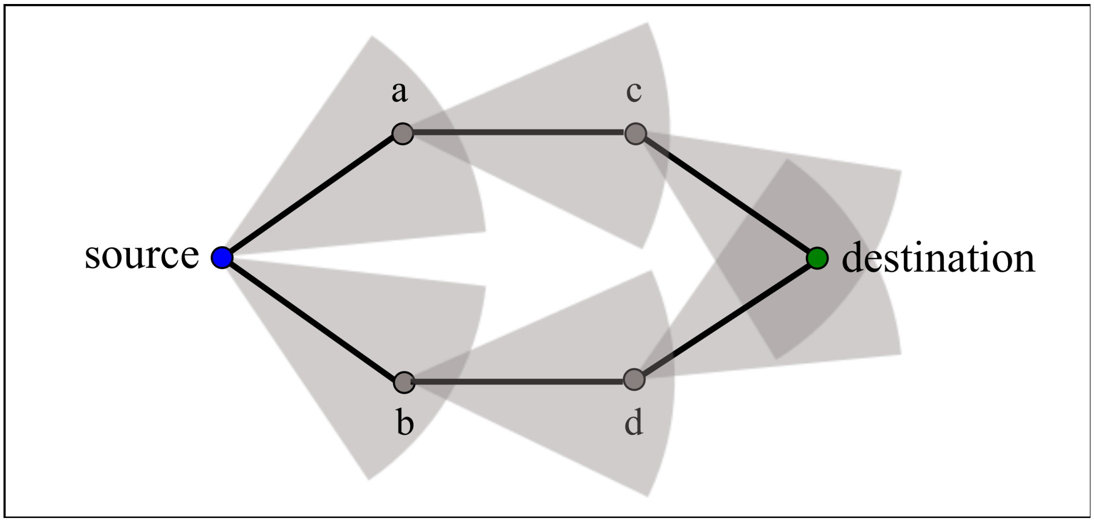


Figure 2.2: MPR-EAPAR

2.2.1.4 MPR-EAPAR

To alleviate the route coupling effect, authors in [47] propose the MPR-EAPAR (Multi-path Routing based on EAPAR). It adopts directional antenna, instead of omni-directional antenna to discover two maximally zone-disjoint and shortest paths. The concept of zone-disjoint means that chosen paths are separated from each other enough to keep the data transmission on one path from interfering with the other path.

As shown in Figure 2.2, the directional antenna is effective in preventing channel contention. The directional antenna, namely EAPAR (Electronically Steerable Passive Array Radiator), is considered to be low-cost, low-power and of a small size; and, it is also used in other related works[49,50]. As we can observe, the direction of the antenna is the key issue for the avoidance of route coupling effect and excessive path length growth. Two types of network-status information, ANLn (active node list) and GLSTn (global link-state table), are maintained by every node inside the network. The ANLn of node n is used to keep track of the communication activities around the entire networks perceived by node n, while the GLSTn is used to keep track of the network topology changes. Both of ANLn and GLSTn are updated by periodical exchanges among neighboring nodes. Based on them, the direction of each node's EAPAR is determined.

Table 2.4: Comparison of multipath protocols

Protocol	Underlying Protocol	Multipath Discovery Metric	Mobility Support	Route Coupling Prevention
SMR [36]	DSR	Node-disjoint	Low	No
AOMDV [43]	AODV	Link-disjoint	Medium	No
HM-AOMDV [37]	AODV	Link-disjoint	Medium	No
SDMR[48]	DSR	Spatial-disjoint	Low	Yes
MPR-EAPAR[47]	AODV	Zone-disjoint	Medium	Yes

This protocol is able to decrease the route coupling effect and the path length growth; but, its ability to have widespread impact is made problematic due to the special kind of directional antennas required. Furthermore, the network-status perception scheme is either very unreliable or very costly, under a highly dynamic scenario, depending on the interval of periodical broadcasting.

2.2.2 Summary

Although the multipath solution has a great number of attractive factors, such as load distribution, it also faces severe challenges. The way to increase path diversity is one challenge that must be tackled. Another issue of multipath solution is the route coupling effect, which is proven to eclipse the advantage of the multipath scheme.

Table 2.4 shows the typical multipath protocols previously explained. As we can observe, few of them consider route coupling effect. Moreover, these works are all based on the traditional route based forwarding scheme, which is certainly not suitable for highly dynamic scenarios. Therefore, a solution which takes both route coupling effect and highly dynamic network topology into consideration is of great demand.

Chapter 3

Underlying Single Path Solution - VIRTUS

In this chapter, the selected underlying single path scheme, namely VIRTUS (Video Reactive Tracking-based Unicast protocol) [15], is introduced. It is a geographic receiver based forwarding solution designed to fulfill the task of video streaming over vehicular networks. According to the performance evaluations, VIRTUS is more capable of video streaming over VANETs than other schemes. There are three main reasons that make it stand out from others. First, it is based on the receiver based forwarding scheme, which provides greater adaptability for the highly mobile nature of VANETs. Second, it incorporates link durability into greedy strategy in order to support continuous video transmission. Third, it proposes innovative forwarding zone, reservation time and a multiple forwarders prevention method to alleviate the drawbacks of a receiver based forwarding scheme. In the following subsections, the protocol is described.

3.1 Streaming Process

Video streaming is initiated by a video requester (i.e., the destination node) by sending a video request to the video sender (i.e., the source node). As soon as the source node hears the request, it starts to transmit video data to the destination. All packets, regardless of whether they are for video content, requests or location information updates, are forwarded by the forwarders (i.e the node responsible for forwarding packet). The forwarder is chosen based on the following process.

The last hop (i.e., previous forwarder) will send a packet to all the neighbors in its communication range by broadcasting. Upon receiving the packet, nodes inside the forwarding zone of the last hop schedule themselves to forward the packet based on the waiting time calculated. Once a node's waiting time expires and it does not overhear any transmission for the same packet, it is considered as the winner of the waiting time competition and is thus elected as the new forwarder. The new forwarder will relay the packet to the next hop level and it will continue to forward the following packets within a calculated time interval called reservation time (λ). When the other nodes inside the forwarding zone of the last hop overhear the transmission of the new forwarder, they will behave differently according to their location relationship with the new forwarder. If the node is located farther from the destination than the new forwarder, it will cancel its own transmission. If the node located closer to the destination than the new forwarder, it will reschedule itself to forward the packet. For such a node, its waiting time calculation should be treated differently depending on whether it is inside the forwarding zone of the new forwarder. If it is inside, it will calculate its waiting time using the location and movement information of the new forwarder. If it is outside the zone, it will calculate its waiting time using the location and movement information of the last hop.

The key design issues of VIRTUS are forwarding zone definition, reservation time estimation and waiting time calculation. They will be described in the following subsections with details. After that, a multiple forwarders prevention scheme is discussed.

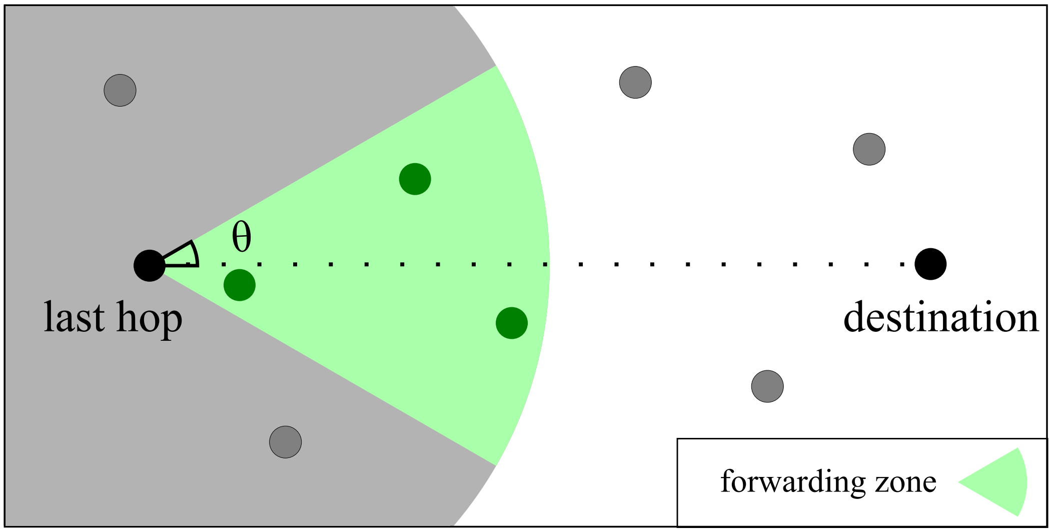


Figure 3.1: Forwarding zone

3.2 Forwarding Zone

One of the major drawbacks of the traditional receiver based solution is that not all nodes are able to overhear the forwarder and cancel their transmission schedules accordingly. This is because the transmission range of the last hop is too large and the new forwarder be able to may not reach all other inner nodes. Therefore, redundant transmissions will happen, leading to very high overhead. Moreover, it will increase the collision rate over the network, raising the delay time while lowering the delivery ratio. In order to solve this problem, VIRTUS introduces the forwarding zone scheme. Only the nodes inside the forwarding zone of the last hop are qualified to become the next forwarder, while other nodes will drop the packets they have received.

As shown in Figure 3.1, the forwarding zone of a last hop can be defined as a sector of its communication circle. The forwarding zone is directed towards the location of the destination node. This directly increases the chance of the nodes close to the destination becoming the next forwarders. The radius of the forwarding zone is fixed as the communication range; thus, the proportion of the zone fully depends on the center angle. As designated in Figure 3.1, VIRTUS chooses half of the center angle, max angle (θ), as the key parameter. If θ is too small, the probability of having suitable forwarding nodes inside the forwarding zone will be decreased. However, if θ is too large, the nodes inside the forwarding zone cannot overhear each other's transmissions and cancel forwarding accordingly. Thus, an appropriate θ should be set according to the traffic topology. Based on analysis and experimental studies, θ is specified as 45º for the highway scenario, and as 90º for the urban scenario.

3.3 Reservation Time

Another dark side of the receiver based scheme is that every forwarder is chosen from a waiting time, which causes an unacceptable end-to-end delay for multi-hop streaming. In video streaming, there are a lot of successive packets within a short time period. Therefore, it is very redundant and time consuming to repeat the waiting time competition for every packet. In order to tackle this problem, VIRTUS proposes the reservation time (λ). The forwarder, who wins the competition for packet transmission, will continue to be forwarder for the following packets within λ. λ stands for the stability of a link between a node and its last hop. The bigger the value is, the more stable the link is. Therefore, λ is estimated based on the future location relationship between a node and its last hop. To be more specific, λ of a node is the minimal value between the maximum reservation time (Λ) and the estimated duration in which the node is residing the last hop's forwarding zone. In order to adapt to the high speed nature of vehicles, Λ will be decreased to a fifth of its original value when two nodes are moving apart (i.e., the angle of two nodes' movement vectors are greater than 90º).

GPS (Global Positioning System) or similar devices are required by all vehicles in order to collect location and movement information. Based on this information, VIRTUS uses Bayesian state estimation [51] to predict a vehicle's future location information. For a node to calculate its reservation time, the location and movement information of the node itself, its last hop, as well as the destination node is needed. The node can directly collect its information from GPS, while other nodes' information needed is piggybacked onto the packet. To be specific, the information concerning the last hop is added by itself, while the information of the destination is piggybacked by the source node. The source is aware of the location and movement information of the destination through the periodical update packets sent by the destination.

3.4 Waiting Time

Waiting time is the major characteristic of a receiver based solution. It directly determines the forwarders selected, thus determining the performance of the solution. Basically, the waiting time depends on the proportions it is composed of as well as the calculation scheme needed to balance these proportions. Apart from most existing greedy solutions which only use geographic advance to calculate the waiting time, VIRTUS calculates its waiting time by not only using geographic advance but by also using link durability in the purpose of choosing the most suitable forwarding nodes.

3.4.1 Geographic Advance

As one proportion of the waiting time, geographic advance (γ_{geo}) is responsible for selecting the nodes closest to the destination. Each node's γ_{geo} is calculated based on the distance between it and the destination ($dis(n_r, n_d)$), as well as the distance between its last hop and the destination ($dis(n_{lh,}n_d)$), which is shown as follows:

$$\gamma_{geo} = 1 - \frac{dis(n_{lh}, n_d) - dis(n_r, n_d)}{r} \tag{3.1}$$

where r is the communication range.

3.4.2 Link Stability

The forwarder is not only responsible for relaying one packet but also for all the following packets' transmissions within its reservation time. Thus, the link durability, named as link stability (γ_{stab}) in VIRTUS, between a forwarder and its last hop should also be considered when calculating the waiting time. The calculation is based on the reservation time (λ) as follows:

$$\gamma_{stab} = 1 - \frac{\lambda}{\Lambda} \tag{3.2}$$

where Λ is the maximum reservation time.

3.4.3 Waiting Time Calculation

In order to adjust the influence of different proportions, namely geographic advance (γ_{geo}) and link stability (γ_{stab}), VIRTUS uses a simple weighting scheme to balance them as follows:

$$\gamma = [\alpha \times \gamma_{stab} + (1 - \alpha) \times \gamma_{geo}] \times \Gamma \tag{3.3}$$

where γ is the waiting time, α is the weighting factor and Γ is the wait time scale.

3.5 Multiple Forwarders Prevention

Although the forwarding zone scheme reduces the possibility of having multiple forwarders in the same hop level, this problem still cannot be totally resolved because different nodes may have similar waiting times. This issue will lead to concurrent transmissions if no prevention method is proposed. In order to solve this problem, VIRTUS simply adds a random delay to each transmission through the lower layers. The delay is randomly chosen between 0 and a maximum random delay. The value of the maximum random delay will significantly impact the performance. If it is too small, the problem of concurrent transmission cannot be solved. If it is too big, the streaming will suffer from high end-to-end delay. According to empirical evaluations, it is set to $10ms$.

If the multiple forwards already exist, they may collide with each other until their reservation time expires. Therefore, VIRTUS proposes a method to overcome this problem. All forwarders will include their waiting times and hop counts onto packets' headers. If a forwarder receives an already transmitted packet, it will first compare the hop count for the received packet with the previously transmitted packet. If the hop counts for both packets are the same, it means the two forwarders are on the same hop level. Then the waiting times associated with both packets will be compared. If the waiting time for the current packet is less than that of the previously transmitted packet, the forwarder will abandon its forwarding duty through canceling its reservation time. Otherwise, no changes will happen and it will continue to relay packets.

3.6 Summary

This chapter introduces the underlying single path solution, VIRTUS, for LIAITHON$^+$. It is a receiver based solution which calculates the waiting time based on geographic information. In order to discover the most suitable intermediate forwarders, the waiting time is based on both the greedy and link durability metric. Moreover, the major problems of adapting a receiver based solution to video streaming are overcome in VIRTUS. First, a forwarding zone scheme is proposed to facilitate forwarder selections. Second, the high end-to-end delay problem caused by unnecessary waiting times is reduced by the reservation time scheme. Third, the multiple forwarders problem caused by similar waiting time is largely avoided. In conclusion, VIRTUS offers a great solution for video streaming, which paves a solid path for LIAITHON$^+$.

Chapter 4

LIAITHON$^+$

This chapter completely describes the proposed work LIAITHON$^+$ (Location-Aware Multipath Video Streaming). This work is trying to discover the most suitable receiver based multipath solution for video streaming over urban VANETs. Because it is hard to find diverse decoupled paths for highway scenarios, an urban scenario is preferred in this work. In order to discover the most suitable path number for video streaming, both two path solution (2-path LIAITHON$^+$) and three path (3-path LIAITHON$^+$) solution using the same model are developed and evaluated. Based on the simulation comparison between 2-path and 3-path solutions, the analysis of the impact on the path number is extended to contain more than three paths. After that, LIAITHON$^+$ is compared with its underlying solution as well as the node-disjoint solution through simulation.

This chapter is structured as follows: Section 4.1 discusses the motivation for the proposed work. Section 4.2 introduces the general video streaming process of LIAITHON$^+$. Section 4.3 describes the core metric for multipath discovery, namely degree of closeness. Based on that, Section 4.4 provides the detailed design concept of a 2-path solution. After that, Section 4.5 describes further the design concept of a 3-path solution. Finally, the simulations and results are elaborated upon in Section 4.6.

4.1 Motivation

Enabling video streaming can either provide an underpinning solution or achieve a higher Quality of Experience (QoE) for applications in VANETs. However, by using vehicles as nodes to transmit video content, a lot of tricky challenges arise from the VANETs' environment. For instance, the video source, destination and intermediate forwarders (i.e the nodes responsible for forwarding packets) may become transient which causes link disconnection. Moreover, these challenges are exacerbated because they face the stringent QoS requirements of video streaming.

VIRTUS, described in Chapter 3, is selected as the underlying single path work for LIAITHON^{+}. By overcoming the major disadvantages of existing receiver based forwarding solutions, VIRTUS achieves the strict QoS requirements of video streaming over VANETs. However, the performance of VIRTUS drops very quickly when the data rate increases. For video streaming, high data rate transmission is an important issue which cannot be omitted. The use of multiple paths to transmit video data simultaneously can significantly alleviate traffic load, which is able to reduce congestions and collisions under high data rate conditions.

Among the approaches used to improve the performance of video transmission over ad hoc networks, multipath scheme is considered as a potential solution. However, the currently most popular multipath schemes reviewed in Chapter 2 are not suitable enough for this task. To our best knowledge, no existing multipath work uses the receiver based forwarding scheme, which is able to adapt to the highly dynamic environment of VANETs. Furthermore, few works have considered the route coupling effect which severely impacts the performance of the multipath scheme.

To our best knowledge, LIAITHON^{+} is the first work to address the task of multipath video streaming over VANETs by using the receiver based forwarding scheme. The core objective of LIAITHON^{+} is to discover the most suitable paths. There are two major differences that set LIAITHON^{+} apart from most of the existing multipath schemes.

First, the traditional route based multipath forwarding schemes (e.g., [35, 36, 38, 43–45]) often use path discovery sessions (e.g., RREQs) to discover multiple suitable paths; in contrast, LIAITHON$^+$ neatly integrates path discovery into the waiting time scheme by using trajectory based strategy. Second, most of the schemes (e.g., [36–38]) focus on finding node-disjoint or link-disjoint paths in order to increase path diversity; while neglecting the increase of path length and route coupling effect between paths. In contrast, LIAITHON$^+$ mainly focuses on discovering relatively short paths with minimal route coupling effect. These characteristics make LIAITHON$^+$ very adaptable to video streaming over vehicular ad hoc networks.

Thus, the motivation of LIAITHON$^+$ can be summarized as follows:

- Reducing the collision and congestion problem under a high data rate.
- Minimizing the route coupling effect.
- Minimizing the path length growth.

4.2 Streaming Process

The streaming process is similar to that of VIRTUS which is described in Section 3.1. However, since the path number is increased from single to multiple, some changes occur accordingly.

4.2.1 Video Stream

Basically, video requester (i.e., the destination node) initiates the video streaming by sending the video requests to the video sender (i.e., the source node). The request packet is comprised of the video identifier, the source node's ID and its position.[1] Unlike the single path solution, the destination node in LIAITHON$^+$ will send several requests

[1] We assume the source node's information is known by the destination node.

one after another to the source node instead of sending only one request. The number of requests are determined by the number of paths, since each request is transmitted, respectively, on each path. This mechanism is used to adapt to the reservation time scheme. It takes much longer for the first packet to reach its destination than the following packets. Because, instead of continuing to relay packets within the reservation time, every forwarder is elected by its waiting time. For a multipath solution, every path has this initialization problem. Therefore, by sending a set of request packets, multiple paths are initialized concurrently during the video requesting session. By this mean, the delay for the following video packets' transmissions can be significantly reduced.

After the source node hears the first request, it will schedule itself to send the video data, which is designated by the video identifier, based on a certain time interval. The video transmission will be triggered if the time interval expires or if the source node receives all the request packets. According to the empirical study, the value of the time interval is assigned to 0.5 s. Under normal situations, all of the requests arrive before the time interval. Thus, this method functions only as a "fuse" for the use of the protocol under extreme conditions, such as in a very sparse area. Once the video is triggered, the source node will take turns transmitting the video packets on each path in order to distribute the traffic load uniformly.

One thing that should be noted in any multipath solution is that packets may arrive at the destination node out of order. In regard to this, we assume that different paths are uniform in terms of packet loss and delay. The paths are composed of the forwarders with the same behavior; moreover, in LIAITHON$^+$, the paths are close enough that can be regarded as they are under the same network topology. Besides that, according to the simulation results, the delay for each frame is significantly lower than the QoS requirement. Thus the disordered packets could be buffered and reordered within an acceptable delay at the destination node.

4.2.2 Path Identifying

For the intermediate nodes to identify the designated path along which a packet is supposed to be transmitted, a special ID (i.e., Path ID) is appended to every packet. To be specific, the source node assigns Path ID into every packet's header, and the value of the ID is iterated in order to uniformly distribute the data load onto different paths. Since the path is composed of forwarders elected from the waiting time in the receiver based forwarding scheme, the receivers calculates different waiting times for packets with different Path IDs.

4.2.3 Forwarder Selection Method

All packets, whether they are for video content, requests or location information updates, are transmitted following the forwarding process described below.

Every forwarder uses broadcasting to send the packet it receives to all neighbors in its communication range. As soon as a node inside the last hop's (i.e., previous forwarder) forwarding zone receives a packet, it checks the Path ID inside the packet's header and use this information to calculate its waiting time. Such a node will schedule itself to forward the packets using the waiting time calculated. Once a node reaches its waiting time and it has not overheard any other transmission for the same packet, it will be automatically elected as the new forwarder. The new forwarder relays the packet as well as the following packets have the same Path ID within its reservation time. If another node inside the forwarding zone of the last hop overhears the transmission, it will check its distance to the destination. If it is farther from the destination than the new forwarder, it will cancel its forwarding schedule. Otherwise, it will check if it is inside the forwarding zone of the new forwarder. If it is, it will reschedule its forwarding based on the location and movement information of the new forwarder. If not, it will reschedule its forwarding based on the location and movement information of the last hop.

In this process, the same forwarding zone (Section 3.2) and reservation time scheme

(Section 3.3) are used. Moreover, the multiple forwarders prevention scheme (Section 3.5) is incorporated into LIAITHON$^+$ as well.

4.2.4 Location and Movement Information

Location and movement information of a node contains the node's ID and position as well as the speed of that node. In order to calculate the waiting time and reservation time, every node in LIAITHON$^+$ is aware of its own location and movement information, that of the source node, the destination node, and its last hop. Information is collected through the following ways. Every node collects its information through the GPS or similar devices installed on it. Every forwarder, including the source, piggybacks its information collected by its GPS onto the packets transmitted. The source node also appends the information of the destination, collected from the periodical location and movement update packet, onto every packet.

4.3 Degree of Closeness

In order to develop an appropriate waiting time scheme for multipath transmission, the metric to assess the quality of the potential forwarders, in terms of multipath transmission, is demanded. This metric is defined as the degree of closeness.

4.3.1 Definition

The multipath qualities of the chosen paths largely determine the performance of a multipath scheme. The set of paths with optimal multipath qualities are able to maximally increase the advantages while decreasing the disadvantages of the multipath scheme. Such a set of optimal paths should satisfy the following conditions. First, they have to be maximally node-disjoint, which ensures enough path diversity for fault tolerance.

Second, the route coupling effect need to be minimized. In other words, the paths should be separate enough to prevent collisions between them. Third, the increase of path length compared to single path solution must be minimized. This effectively controls the increase of delay and overhead caused by the increase of hop number. As we discussed previously, there exists a trade-off between the second condition which keeps paths as separate from each other as possible and the third condition which keeps paths as close to the shortest path as possible. Thus, a set of optimal paths should also be able to balance this trade-off in the best way.

By using optimal paths as references, multiple paths are selected based on their respective closeness to the optimal paths. Since each path is composed of a set of forwarders, choosing an appropriate path is equal to choosing a set of appropriate forwarders. In LIAITHON$^+$, forwarders are elected by the waiting times they calculated. Therefore, in order to calculate the appropriate waiting time, the metric used to measure the capability of a receiver in terms of its closeness to the optimal path is required. This metric is defined as the degree of closeness (γ_{doc}). Any intermediate node's γ_{doc} is calculated based on the difference between the node's separation factor (μ) and the optimal path's optimal separation factor (μ_{opm}). In the following subsections, μ and the μ_{opm} will be elaborated upon.

4.3.2 Separation Factor

The separation factor (μ) of a node can be regarded as a signed value. As depicted in Figure 4.1, we assume the source node is the origin of a quadrant and the destination node lies on the positive x-axis. Absolute value of μ equals to the distance from the intermediate node to the x-axis. The sign of μ is determined by the relative location of the intermediate node. If this node is located in the first or second quadrant, such as intermediate node A, its μ is positive. If this node is located in the third or fourth quadrant, such as intermediate node B, its μ is defined as negative.

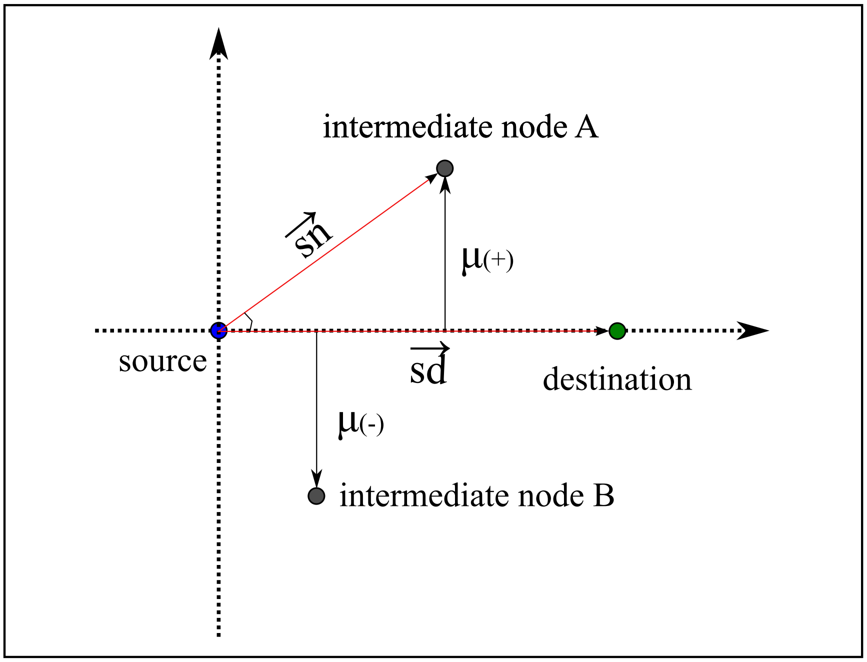


Figure 4.1: Separation factor

A node can calculate its separation factor based on its location information collected from GPS, as well as the location information of the source node and the destination node collected from the packet's header. The following example shows how to calculate the μ of the intermediate node A, depicted in Figure 4.1. First, the vector from the source node to the destination node, $\vec{sd}$ (assumed as $(a_x, a_y, 0)$), is calculated using the location of the source and the destination. We then calculate the vector from the source node to node A, $\vec{sn}$ (assumed as $(b_x, b_y, 0)$). Based on the two vectors, μ can be calculated as follows:

$$\mu = \frac{(a_x \times b_y - a_y \times b_x)}{\sqrt{(a_x)^2 + (a_y)^2}} \tag{4.1}$$

where the numerator refers to the z-component of $\vec{sd} \times \vec{sn}$ and the denominator refers to the length of $\vec{sd}$.

4.3.3 Optimal Separation Factor

In LIAITHON^{+}, most forwarders on an optimal path have the same separation factor. Such a μ is defined as the optimal separation factor (μ_{opm}) for the optimal path. The specific selection of optimal paths and their μ_{opm} depends on the number of paths, which will be described in Section 4.4 and Section 4.5.

4.4 LIAITHON^{+} with Two Paths

In this section, 2-path video streaming solution is proposed on the basis of the streaming process (Section 4.2) and the concept of the degree of closeness (Section 4.3). The major focus is to design an appropriate degree of closeness calculation scheme and a suitable waiting time calculation method for multipath transmission. In addition, two route coupling prevention schemes are used to further avoid collisions between the two selected paths.

4.4.1 Degree of Closeness

To calculate the degree of closeness, we have to select the optimal pair of paths as well as define their optimal separation factors. Also, a method of calculating the difference between the separation factor and optimal separation factor needs to be developed.

4.4.1.1 Optimal Pair of Paths

Taking consideration the conditions set for the optimal paths described in Subsection 4.3.1, the optimal pair of paths are defined as path 1 and path 2 as shown in Figure 4.2. If we disregard the short path segments originating from the source and arriving at the destination (i.e., segments a, b, c and d), both of the paths run parallel to the dotted line crossing the source and the destination. This means that most forwarders of an

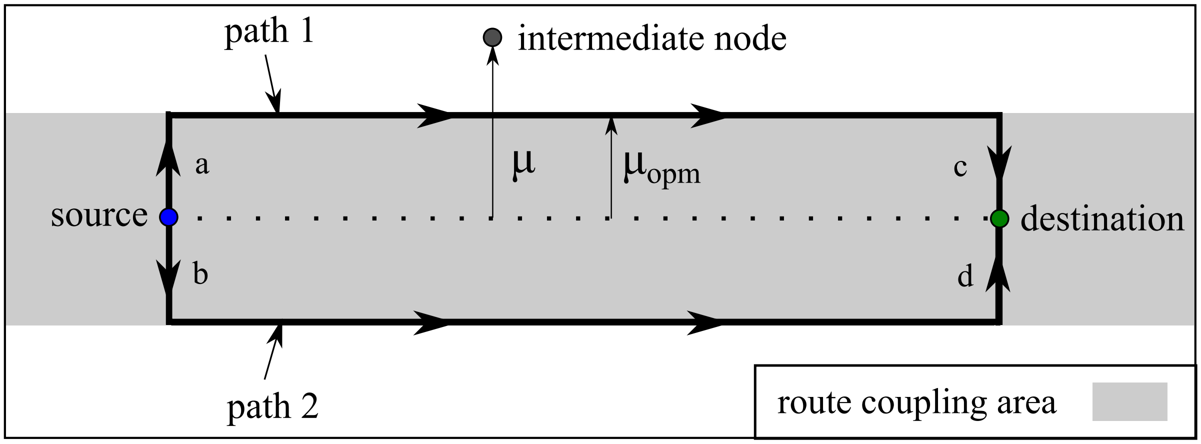


Figure 4.2: Optimal pair of paths

optimal path have the same separation factor (μ). According to the definition of optimal separation factor (μ_{opm}) in Subsection 4.3.3, this μ is selected as the μ_{opm} for the optimal path. Moreover, the two paths are symmetrical about the dotted line crossing the source and the destination. Therefore, they have the same absolute value but opposite signs of μ_{opm}.

As described in Subsection 4.3.1, the optimal paths should balance the trade-off between avoiding the route coupling effect and controlling path length growth. Since the distance between paths is determined by the absolute value of μ_{opm}, the trade-off can be balanced through this value. If the μ_{opm} is too small, the two paths will be too close to each other, which may incur route coupling between them. However, an excessively large μ_{opm} will result in a significant increase in path length. We chose the transmission range[2] as the absolute value for μ_{opm}, which prevents intermediate nodes on one optimal path from overhearing transmissions on another path. Moreover, there is no hidden node between two optimal parallel paths. In other words, any node which overhears the transmission on one path cannot overhear the transmission on the other path at the same time. Therefore, even if some forwarding nodes drop into the route

[2]We have made two reasonable assumptions: first, that the transmission range is the same as the interference range and secondly, that all the nodes in the network have the same transmission range.

coupling area indicated in Figure 4.2, they will not get involved in severe collisions with the nodes on the other path. As for the path length, just a few hops have been increased when compared to the shortest path between the source and the destination (i.e., the dotted line).

Based on the illustrations above, the μ_{opm} for the two optimal paths are set as follows:

$$\begin{cases} \mu_{opm} = TR, & path\,1 \\ \mu_{opm} = -TR, & path\,2 \end{cases} \tag{4.2}$$

where TR is the transmission range. One thing that should be noted is that these equations are only used for video data packets as well as location and movement update packets traveling from the source to the destination. In order to maintain consistency, the request packets as well as location and movement update packets on the reverse direction (i.e., destination to source) will be using the reverse equations (i.e., μ_{opm} is set as TR for path 2 and $-TR$ for path 1).

4.4.1.2 Degree of Closeness Calculation

In order to facilitate the waiting time scheme, we define that the smaller the numerical value of γ_{doc} is, the closer the node is to the optimal path. If γ_{doc} of a node equals 0, it means this node is on the optimal path. γ_{doc} is calculated based on the difference between μ and μ_{opm}. The equation is as follows:

$$\begin{cases} \gamma_{doc} = \frac{\mu - \mu_{opm}}{\mu_{opm}}, & \left|\mu\right| \geq \left|\mu_{opm}\right| \\ \gamma_{doc} = \frac{\mu_{opm} - \mu}{\mu_{opm}} \times pr, & \left|\mu\right| < \left|\mu_{opm}\right| \end{cases} \tag{4.3}$$

where μ_{opm} is selected based on Equation 4.2 by identifying the Path ID in the packet's header. pr is the penalty ratio and its value is greater than 1. If a node's separation factor is numerically smaller than the optimal separation factor, this indicates that the node falls inside the route coupling area depicted in Figure 4.2. Therefore, in the case of such a node, we put a penalty *pr* on its degree of closeness to reduce its chance of being the next forwarder.

4.4.2 Area-aware Waiting Time Calculation

The waiting time for LIAITHON$^+$ is not only based on geographic advance and link stability described in VIRTUS, but is also based on the degree of closeness. In other words, if a node is on the optimal path, it is more likely to forward the packet. Moreover, in order to keep the chosen pair of paths closer to the optimal pair of paths shown in Figure 4.2, LIAITHON$^+$ uses a location-aware waiting time calculation scheme; in this scheme, the waiting time calculation method varies according to the different areas an intermediate node is located in.

We divide the whole network area into source area, destination area and intermediary area (i.e., the remaining area). The main reason behind this classification is that transmissions close to the destination and source are highly prone to collision when compared to the intermediary area. Furthermore, the source and destination areas should be treated separately due to the different problems they have. For the source area, the two chosen paths may not split away from each other immediately. For the destination area, paths may merge too early or too late. The different waiting time calculation methods are elaborated in the rest of this subsection.

4.4.2.1 Intermediary Area

If a node lies outside both the source area and the destination area, it is considered to be within the intermediary area. In this area, the distance between the two optimal paths is two times that of the transmission range. If one forwarder is far away from the optimal path, the performance will not be affected significantly. However, a series of forwarders far away from the optimal path will have a serious impact on the performance. Such a series of forwarders inside the route coupling area will cause severe collisions, while such a series of forwarders outside the route coupling area will lead to a significant increase in path length. In order to prevent this problem, the following process is performed. Every forwarding node appends the value of its degree of closeness (γ_{doc}) onto the packet it transmitted. A node will check the last hop's γ_{doc} upon receiving a packet. If the value is less than a predefined constant, namely the closeness threshold (σ), it means the last hop is close to the optimal path. The node will then use the following equation to calculate the waiting time (γ):

$$\gamma = [\alpha \times \gamma_{stab} + \beta \times \gamma_{doc} + (1 - \alpha - \beta) \times \gamma_{geo}] \times \Gamma \tag{4.4}$$

where α and β are weighting factors and Γ is the waiting time scale. γ_{geo} and γ_{stab} are, respectively, the geographic advance (Subsection 3.4.1) and the link stability (Subsection 3.4.2) introduced in VIRTUS. If the value of the last hop's γ_{doc} is greater than σ, it means that the last hop is far away from the optimal path. Thus, the next forwarding node should be as close to the optimal path as possible. Under this circumstance, the waiting time is calculated as follows:

$$\gamma = [\alpha \times \gamma_{stab} + \beta \times \gamma_{doc}] \times \Gamma \tag{4.5}$$

γ_{geo} and γ_{doc} both select the forwarder by its location, but they play different roles. γ_{geo} is responsible for choosing forwarders which are close to the destination, while γ_{doc} chooses forwarders which are close to the optimal path. Therefore, we eliminate the influence of γ_{geo} in order to enhance the influence of γ_{doc}.

4.4.2.2 Source Area

If a node is one hop away from the source node, this node is regarded as within the source area. In order to split two paths immediately, the max angle (θ) for the forwarding zone of the source is set to 180º. The situation for the source area is similar to the situation in which the last node is far away from the optimal path; thus, the same equation (i.e.,

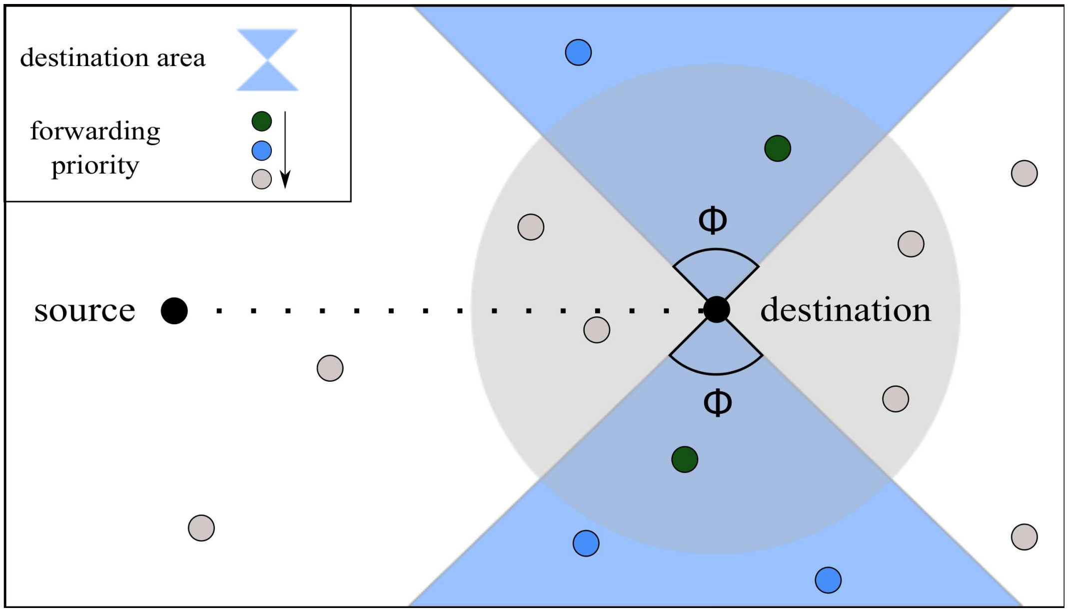


Figure 4.3: Destination area

Equation 4.5) for waiting time calculation is followed.

4.4.2.3 Destination Area

As depicted in the blue shading of Figure 4.3, the destination area is symmetrically located above and blew the dotted line running between both the source and the destination. The coverage area of the destination area is determined by the destination angle (ϕ). Every node is aware of its own location information and that of the destination node as well as that of the source node, thus, the vector from an intermediate node to the destination (assumed as $\vec{nd}$) and the vector from the source to the destination (assumed as $\vec{sd}$) can be calculated. A node can easily check if it is inside the destination area by using the dot product of $\vec{nd}$ and $\vec{sd}$. If and only if the node satisfies the following condition statement, it can be regarded as within the destination area:

$$\frac{\pi - \phi}{2} \leq \arccos(\frac{\vec{nd} \cdot \vec{sd}}{\left|\vec{nd}\right| \times \left|\vec{sd}\right|}) \leq \frac{\pi + \phi}{2} \tag{4.6}$$

If the forwarding nodes are inside the destination area, the two paths will merge together in an appropriate angle instead of merging too early or too late. This can significantly reduce collisions among transmissions near the destination. Therefore, the nodes inside the destination area take half of the waiting time scale (Γ), to gain advantage in the forwarding competition against exterior nodes (i.e., gray nodes). The waiting time is calculated as follows:

$$\gamma = [\alpha \times \gamma_{stab} + \beta \times \gamma_{doc} + (1 - \alpha - \beta) \times \gamma_{geo}] \times \frac{\Gamma}{2} \tag{4.7}$$

Furthermore, if a node is located inside the destination area as well as the destination's communication range (i.e., green nodes), it has the highest priority to forward packets. Geographic advance is not important for these nodes, since they are one hop away from the destination. Thus, they schedule themselves to forward packets based on their γ_{stab} and γ_{doc} only:

$$\gamma = [\alpha \times \gamma_{stab} + \beta \times \gamma_{doc}] \times \frac{\Gamma}{2} \tag{4.8}$$

Additionally, the μ_{opm} for nodes within the destination area drops into half of the transmission range. By this method, the relaying nodes are more likely to lie inside the destination's communication range and they can forward packets to the destination directly.

To put it simply, the destination area scheme classifies nodes into hierarchical system in terms of their suitability for becoming the forwarder. Moreover, it does not prevent the nodes outside the destination area from forwarding packets in case the inner nodes do not exist.

4.4.3 Route Coupling Prevention

In co-operation with the waiting time scheme stated above, route coupling prevention schemes are proposed to further reduce the potential collision problem between paths.

4.4.3.1 Keep Away From the Opposite Side

By dividing the whole area into two sides, with a line that connects both the source and the destination, the two optimal paths are positioned on opposite sides (as we can see in Figure 4.2). A packet can only be transmitted by nodes on the same side of its optimal path. This ensures that the two chosen paths are always on the opposite sides, which prevents them from twisting together while ensuring that the chosen paths totally node-disjoint. Once a node receives a packet, it will check the optimal separation factor from the Path ID and will then compare it to its own separation factor. If they have the same sign, the node will schedule itself to forward the packet. If not, it will discard the packet immediately.

4.4.3.2 Overhear and Cancel

If a forwarder overhears packets from the last hop which is on the opposite side, it means the forwarder is, to some extent, prone to collide with forwarders on the other side. Therefore, such forwarder will cancel its reservation time in order to leave the chance for other suitable node to win the forwarding competition for the next packet.

4.5 LIAITHON$^+$ with Three Paths

Based on the two path streaming solution discussed in the last section, we further exploit a 3-path solution in order to discover the impact on the path number growth. This scheme is using the same streaming process (Section 4.2) and same concept of degree of closeness (Section 4.3). The degree of closeness calculation scheme and the waiting time calculation scheme from the 2-path solution are adopted accordingly with some changes. Moreover, a new route coupling prevention scheme is developed specifically for three paths streaming.

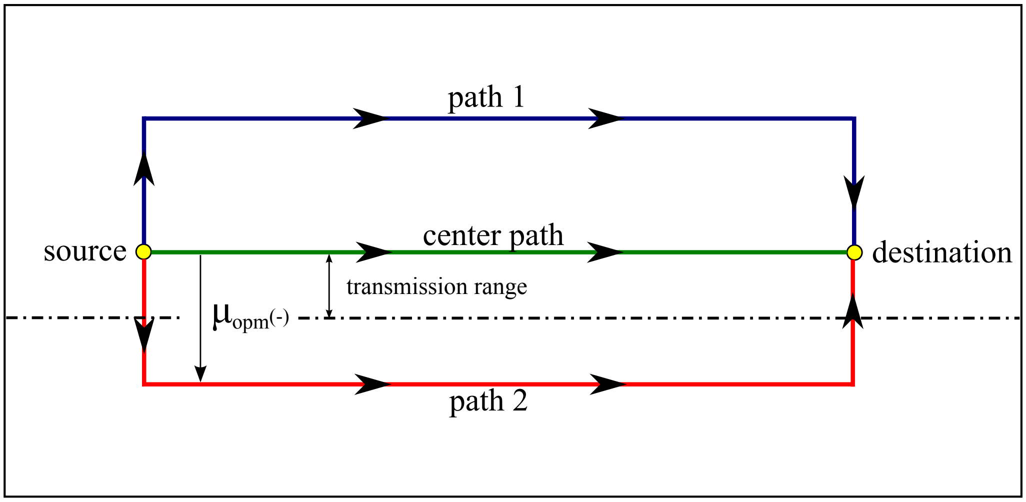


Figure 4.4: Optimal set of paths

4.5.1 Degree of Closeness

We first define the optimal set of paths and assign the optimal separation factors for each path. Using the optimal separation factors as reference, the degree of closeness calculation method is developed.

4.5.1.1 Optimal Set of Paths

The three optimal paths are shown in Figure 4.4 as path 1, path 2 and the center path. The center path is the straight line running from the source node to the destination node. All forwarders of the center path have the same separation factor: 0. Thus, we define it as the optimal separation factor (μ_{opm}) for the center path. As we can observe, the side paths (i.e., path 1 and path 2) are almost the same as the optimal pair of paths in 2-path solution. The only difference is the absolute value of μ_{opm} is increased, in order to leave enough space for the center path. We chose two times of the transmission range as the absolute value of their μ_{opm}. Thus, the distance between any two optimal paths is long enough to prevent route coupling as well as hidden nodes between them. Moreover, the length of the side path is only a few hops more than that of the 2-path solution.

In conclusion, the μ_{opm} for three optimal paths are set as follows:

$$\begin{cases} \mu_{opm} = 2 \times TR, & path\,1 \\ \mu_{opm} = -2 \times TR, & path\,2 \\ \mu_{opm} = 0, & center\,path \end{cases} \tag{4.9}$$

where TR is the transmission range. Like the 2-path solution, packets in the reverse direction will use reverse equations (i.e., μ_{opm} is $2 \times TR$ for path 2 and $-2 \times TR$ for path 1).

4.5.1.2 Degree of Closeness Calculation

Once a node receives a packet, it will check the Path ID associated within the packet's header. If the Path ID refers to the side path, the degree of closeness (γ_{doc}) calculation is the same as the one described in 2-path solution (Equation 4.3). If the Path ID indicates the center path, the degree of closeness is calculated as follows:

$$\begin{cases} \gamma_{doc} = \frac{\left|\mu\right|}{TR}, & \left|\mu\right| \leq \frac{TR}{2} \\ \gamma_{doc} = \frac{\left|\mu\right|}{TR} \times pr, & \left|\mu\right| > \frac{TR}{2} \end{cases} \tag{4.10}$$

where pr and TR are the penalty ratio and the transmission range, respectively.

4.5.2 Area-aware Waiting Time Calculation

Like 2-path solution, the waiting time calculation is determined by the area in which the intermediate node is located. Since the side paths resemble highly the optimal pair of paths in the 2-path solution, a node calculates its waiting time for a packet, with Path ID refers to the side path, using the same calculation scheme described in Subsection 4.4.2.

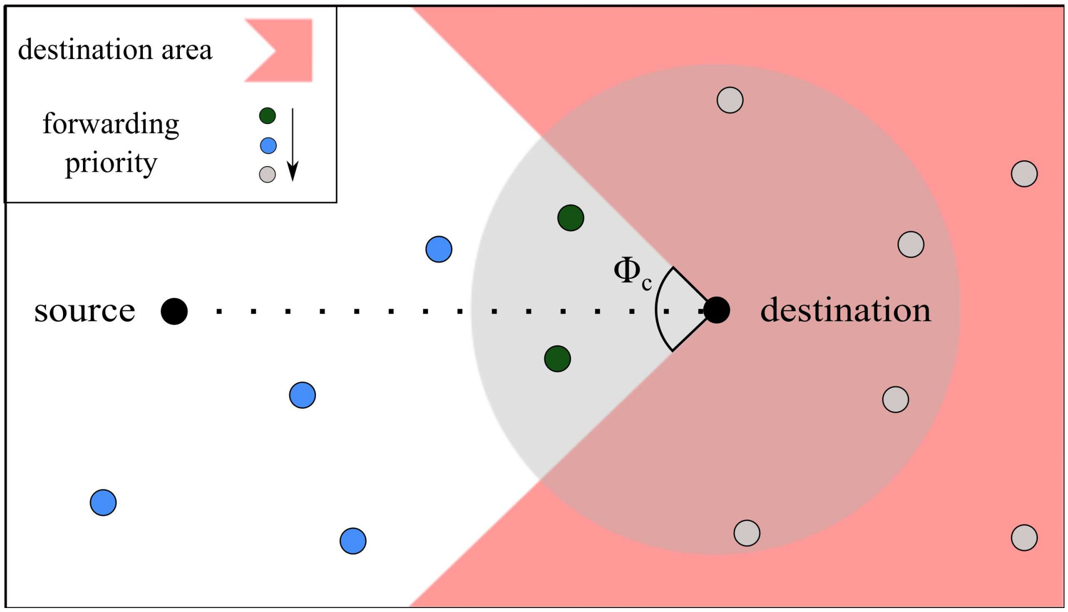


Figure 4.5: Destination area for center path

4.5.2.1 Center Path

As for Path ID indicating the center path, there is no change applied to the waiting time calculation scheme for nodes inside the intermediary area. While the nodes inside the source area keep the same waiting time calculation scheme, except that the max angle (θ) for the source node's forwarding zone remains set as 90º instead of 180º. This effectively reduces collisions between the side path and the center path around the source area, since the forwarders for the center path are, to some extent, geographically separated from the forwarders on the side path. However, the destination area scheme is totally different, because the major issue now is to keep the forwarders as close to the center line as possible.

Figure 4.5 illustrates the destination area for the center path. Since the optimal paths are symmetrical about the dotted line crossing both the source and the destination, the destination area complies to symmetries as well. Its coverage is determined by the center destination angle (ϕ_c). A node is regarded as inside the destination area if and only if it satisfies the following condition statement:

$$\arccos\left(\frac{\vec{nd} \cdot \vec{sd}}{\left|\vec{nd}\right| \times \left|\vec{sd}\right|}\right) \leq \frac{\phi_c}{2} \tag{4.11}$$

where $\vec{nd}$ and $\vec{sd}$ are, respectively, the vector from the intermediate node to the destination node and the vector from the source node to the destination node.

If the forwarding nodes are inside the destination area (i.e., gray nodes), the center path is more likely to collide with the side paths. In order to reduce the chance of having forwarders inside it, the waiting time scale (Γ) for such inside nodes will be doubled. Therefore, waiting time (γ) is calculated as follows:

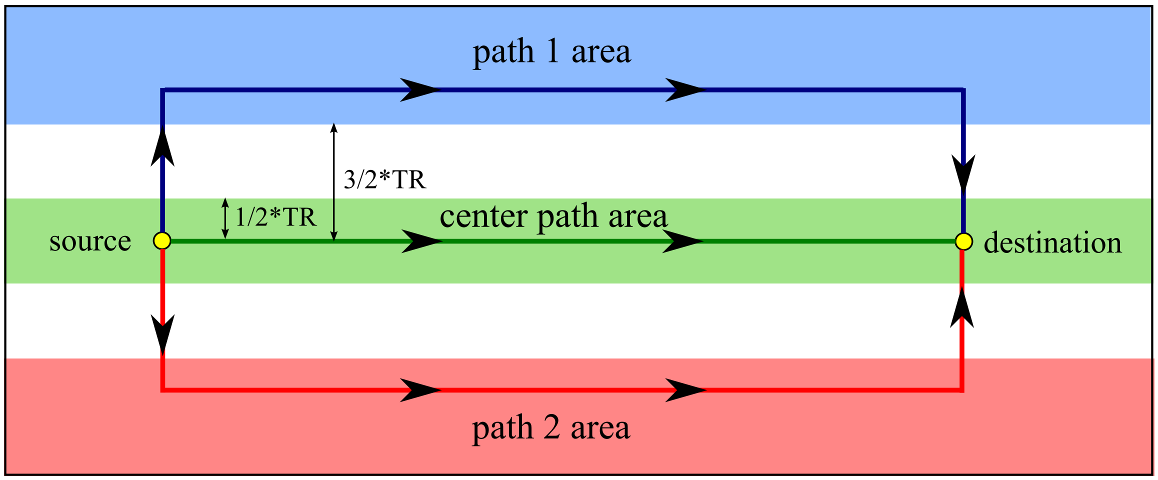
$$\gamma = [\alpha \times \gamma_{stab} + \beta \times \gamma_{doc} + (1 - \alpha - \beta) \times \gamma_{geo}] \times \Gamma \times 2 \tag{4.12}$$

where α and β are weighting factors and γ_{stab}, γ_{doc}, γ_{geo} are link stability, degree of closeness and geographic advance. If the forwarding nodes are located inside the transmission range of the destination node while outside the destination area (i.e., green nodes), they are treated as the highest priority nodes for the forwarding competition. Because γ_{geo} should not be concerns for such nodes, we remove this component accordingly:

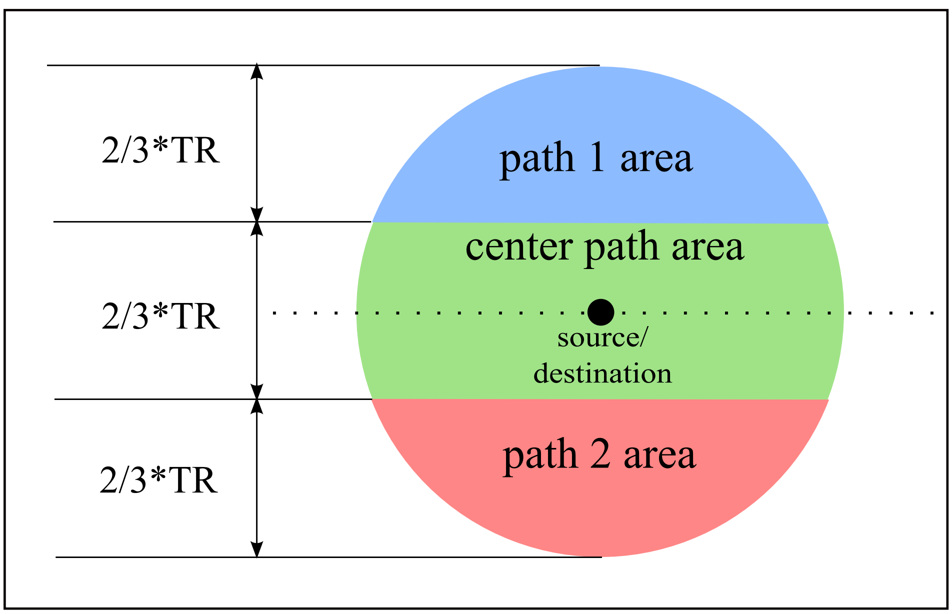
$$\gamma = [\alpha \times \gamma_{stab} + \beta \times \gamma_{doc}] \times \Gamma \tag{4.13}$$

4.5.3 Route Coupling Prevention

Along with the path number growth, the route coupling effect has been raised as well. In order to prevent path 1 and path 2 from being twisted together, one of the 2-path route coupling prevention schemes (Subsection 4.4.3.1) is adopted. However, the route coupling prevention scheme cannot only emphasize collisions between path 1 and path 2; collisions between side paths and the center path should be actually paid more attention.



(a) Regular situation



(b) One hop to source/destination

Figure 4.6: Optimal forwarding area

We first define three optimal forwarding areas for path 1, path 2 and the center path. They are depicted in Figure 4.6 (a) in blue, green and red respectively. If the chosen paths are completely inside their own optimal area, no route coupling will occur among them. TR refers to the transmission range. The center path area should be kept a transmission range away from the side path area. Thus, by evenly dividing the remaining distance, the edge of the center path area and the side path area are set to $\frac{1}{2} \times TR$ and $\frac{3}{2} \times TR$.

However, the normal optimal forwarding area cannot be applied to the forwarders which are one hop away from the source or the destination. For such forwarders, we just simply divide the transmission circle into three forwarding areas each with a height of $\frac{2}{3} \times TR$ as shown in Figure 4.6 (b).

Based on the defined optimal forwarding area, the route coupling prevention scheme is used as follows. If a forwarder on one path overhears a transmission from the forwarder on another path, this forwarder has the possibility of colliding with other forwarders. Thus it needs to check if it is inside its optimal forwarding area by using its separation factor (μ). If it is outside its forwarding area, it will leave its forwarding position by canceling its reservation time. Otherwise, it will keep forwarding during the reservation time calculated.

4.6 Simulations and Results

Through implementations, 2-path and 3-path LIAITHON$^+$ are validated by multipath graphs and compared by using simulation results. Furthermore, in order to evaluate the performance of LIAITHON$^+$, VIRTUS and a node-disjoint solution are used in comparison.

Table 4.1: Simulation Parameters

Parameter	Value
Radio Propagation Model	Two Ray Ground
Mac Layer	IEEE 802.11
Radio Range	300m
Antenna	Omni Antenna

Table 4.2: Mobility Parameters

Parameter	Value
Number of Vehicles	1000
Area Size	3500m * 3500m
Segment Length	100m
Speed Range	5 to 30 m/s
Source Location (x,y)	(650,650)
Destination Location (x,y)	(2850,2850)

4.6.1 Simulation Environment

The simulation is conducted under NS2 (Network simulator 2, version 2.34) [52]. The simulation parameters are set according to Table 4.1. The mobility model we used is Urban Mobility Model (UMM) [53], which is an extension of Bai's Manhattan Model. The parameters for mobility model are specified in Table 4.2. The size of the area and the location of the source node and the destination node are selected in order to keep enough distance between the source and the destination as well as to ensure that the side paths will not reach the edge of the area. When applying the proposed solution to a real time scenario, we assume the urban area is big enough to cover the two different paths.

In order to provide more realistic simulation results, we have conducted the simulation by transmitting the real video content. The content we used is a clip of MPEG video (akiyo_cif) with a resolution of 486×360. We used EvalVid — A Video Quality Evaluation Tool-set [54] to monitor the frame loss and delay. The video data is fitted into 353 packets (1000 bytes payload for each packet).

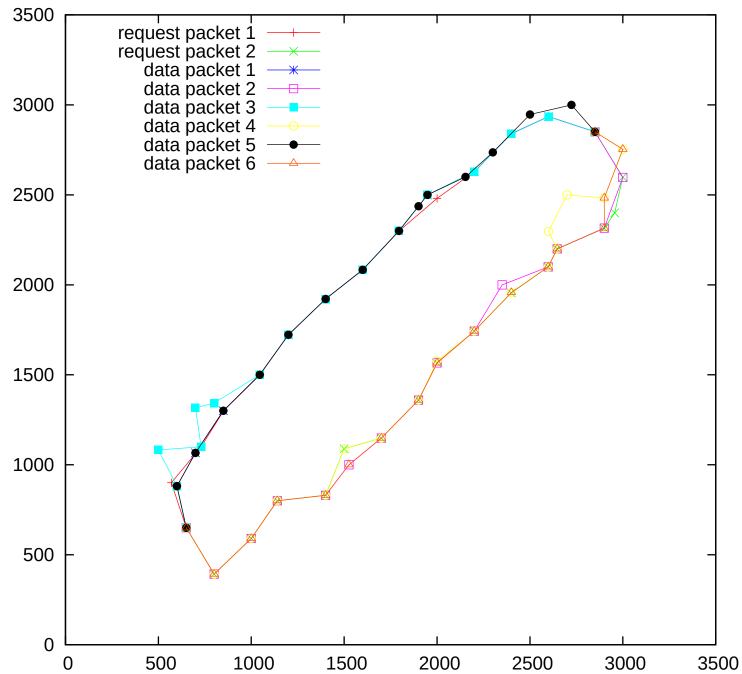
Tool R [55] is used to plot the performance graphs as well as to calculate the confidence interval. The confidence interval is calculated based on Student's t distribution [56] with a 95% confidence level. Every plotted result, in the graphs, is the average of the results generated from 10 to 20 independent simulations, each under a different instance of the same mobility model.

4.6.2 Multipath Graph

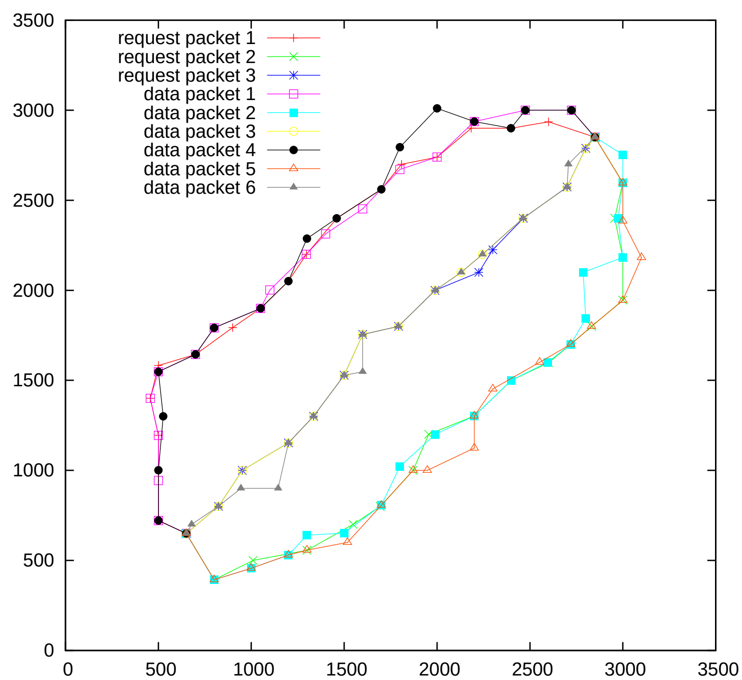
In order to validate if the chosen forwarders are stick to their optimal paths, the most direct and interesting way is to track the paths and plot them into a multipath graph. Every forwarder appends its node ID and position coordinates onto the packet's header. Once the destination or the source receives a packet which has not been received before, it will print all the forwarders' location information. These position coordinates are then plotted into multipath graphs by Gnuplot [57]. Figure 4.7 has shown the multipath graph for 2-path and 3-path solution with total requests as well as six video data packets[3] at a data rate of 2000kbps. For both schemes, the plotted paths highly correspond to the optimal paths. Moreover, as can be measured from the graph, the distance between any two neighboring paths is within the expectation.

Although the multipath graph cannot be used to precisely evaluate the performance, it provides a direct insight on the protocol's behavior as well as offering a method to target the potential problem. Another use of the multipath graph is to adjust the multipath parameters, such as penalty ratio (pr) and closeness threshold (σ), in a way best for performance.

[3] We only plot the multipath graph with few packets, since it is clearer for analysis due to less collisions and congestions.



(a) 2-path



(b) 3-path

Figure 4.7: Multipath graph

Table 4.3: LIAITHON$^+$ Parameters

Parameter	Value	Parameter	Value
Weighting factor (α)	40%	Closeness threshold (σ)	0.5
Weighting factor (β)	30%	Max angle (θ)	90º
Waiting time scale (Γ)	100ms	Destination angle (ϕ)	45º
Max reservation time (Λ)	5s	Center destination angle (ϕ_c)	45º
Penalty ratio (pr)	2		

4.6.3 Discovering the Appropriate Path Number

In order to discover the proper path number for video streaming over urban VANETs, we have implemented both two path and three path mechanisms. In this subsection, we compare the two schemes in terms of frame loss, delay and cost.

4.6.3.1 Parameter Settings

Table 4.3 shows the parameters set for both of the protocols. The waiting time scale (Γ) and max reservation time (Λ) are inherited from VIRTUS; thus, they are set according to the original work [15]. A high way scenario can be simulated as a curve with a large curvature; therefore, a relatively small max angle (θ) can cover most preceding vehicles in the forwarding zone. However, for the grid-like city scenario, a small θ is not suitable anymore. Therefore, through many experiments, we increased the θ by doubling it. As for weighting factors (α and β) and multipath parameters (i.e., penalty ratio (pr), closeness threshold (σ), destination angle (ϕ) and center destination angle (ϕ_c)), they are chosen in the best combinations according to abundant multipath graph analysis and exhaustive experimental evaluations.

4.6.3.2 Performance Comparison

As we can observe from Figure 4.8, in terms of frame loss, the 3-path solution is slightly worse than the 2-path solution at 500kbps or lower. This is mainly because the packet loss probability on side path has increased due to the path length growth. By dividing

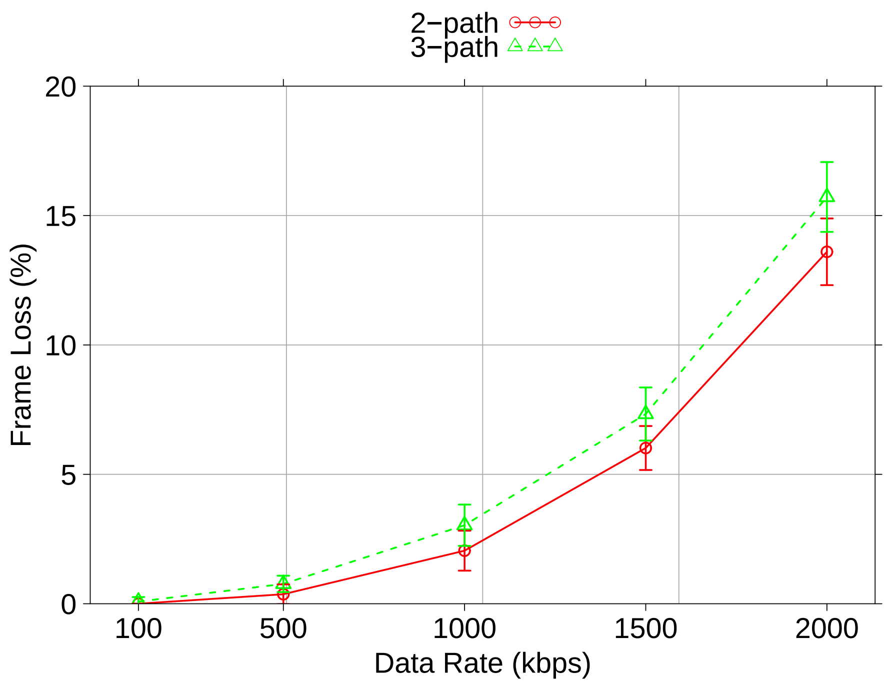


Figure 4.8: 2-path and 3-path LIAITHON^{+} comparison: Frame Loss

the traffic load into three instead of two paths, the 3-path solution should have a lower frame loss rate under a intense traffic load condition (i.e., a higher data rate condition). However, when the data rate goes higher, the 3-path solution becomes even worse than the 2-path solution. This reveals that the benefits of load distribution are eclipsed by the increase of route coupling effect and the growth of path length. By introducing the center path in between two side paths, the trade-off between controlling the route coupling effect and decreasing the path length is stressed to the maximum. It is almost impossible to prevent collisions between paths with a reasonable path length increase. Moreover, the bottleneck, which are severe collisions around the source and the destination, cannot be prevented by a higher degree of load distribution. Additionally, compared to 2-path solution, an additional path introduces more forwarders. They are crowded in the network, especially around the source node and the destination node, contending for the channel access.

Delay is calculated from the average delay of each frame. Figure 4.9 indicates that

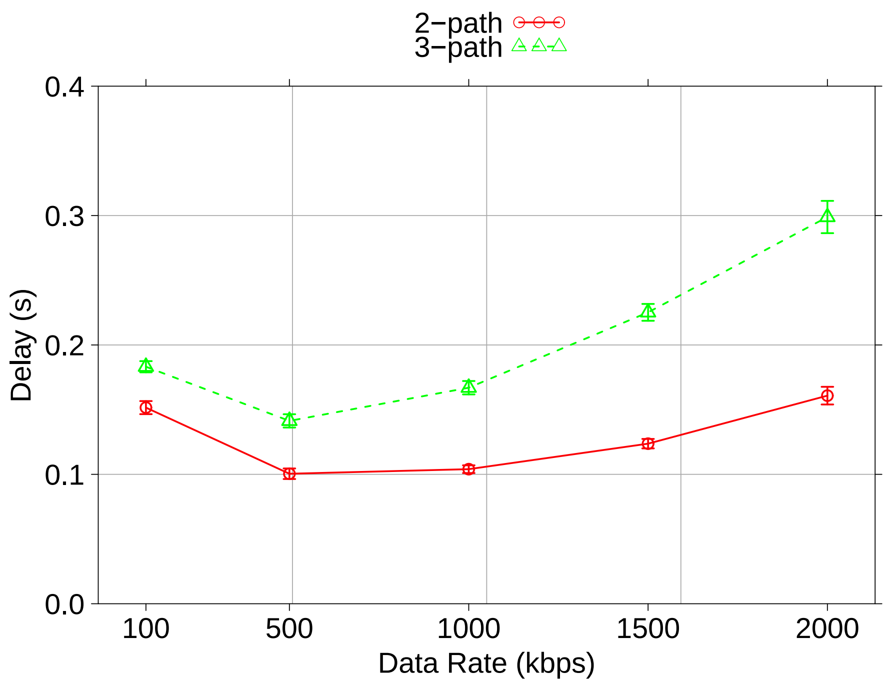


Figure 4.9: 2-path and 3-path LIAITHON$^+$ comparison: Delay

the delay of the 3-path solution is higher than that of the 2-path solution at any data rate. The main reason is that the increase of the side paths' lengths leads to a higher number of hop. Moreover, as data rate goes up, the route coupling effect is increased with a higher rate in 3-path solution. As a result, the delay increase rate in 3-path solution is greater than that of 2-path solution. Another interesting observation is that the delay at 100kbps is not the lowest one for both schemes. This is because, at a very low data rate, a larger percentage of forwarding nodes are chosen based on the waiting time instead of the reservation time.

Figure 4.10 shows the cost comparison. The cost is measured using the ratio of the number of total transmissions to the number of packets sent. The cost of the 3-path solution is higher than that of the 2-path solution, because of the path length increase problem and route coupling problem stated previously.

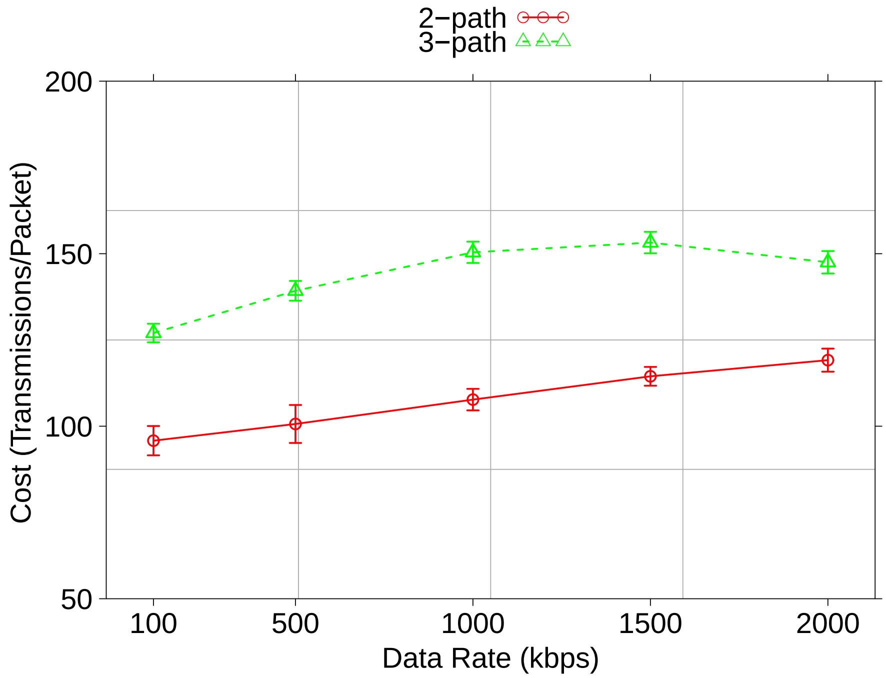


Figure 4.10: 2-path and 3-path LIAITHON$^+$ comparison: Cost

4.6.3.3 Summary

The comparison shows that the 2-path solution outperforms the 3-path solution in every aspect of performance. The advantage of adding an additional path is to further distribute the heavy traffic load under a higher data rate and to add path diversity to increase fault tolerance. However, according to the performance comparison and analysis between 2-path and 3-path solutions, these advantages are totally eclipsed by disadvantages which are the growth of path length and the increase of the route coupling effect. It is reasonable to deduce that this problem will get even worse when the path number becomes greater than three. Therefore, we can conclude that the appropriate path number for LIAITHON$^+$ is two.

Table 4.4: VIRTUS Parameters

Parameter	Value	Parameter	Value
Weighting factor α	50%	Max reservation time (Λ)	5s
Waiting time scale (Γ)	100ms	Max angle (θ)	90º

4.6.4 Performance Evaluation

Last subsection has selected the optimal path number for LIAITHON$^+$ as two. In this subsection, we further compare the 2-path LIAITHON$^+$ with two other protocols in order to evaluate its performance. When we mention LIAITHON$^+$ in this subsection, it refers to 2-path LIAITHON$^+$ only.

4.6.4.1 Single Path Solution

In order to demonstrate the performance enhancement brought about by load distribution, the underlying single path solution, VIRTUS, is used for comparison. Table 4.4 shows the parameters for VIRTUS, they are set in a way intended to maximize VIRTUS's performance according to the original work [15]. According to the performance evaluation of VIRTUS, it has already achieved a great performance when compared to regular geographic receiver based solution as well as gossiping solution.

4.6.4.2 Node-Disjoint Solution

As we have reviewed in Chapter 2, most of the multipath works focus on discovering a set of disjoint paths. Thus it is necessary to compare LIAITHON$^+$ with a node-disjoint multipath solution, which is based on the receiver based forwarding scheme, for the purpose of investigating the performance gain with route coupling effect and path length growth taken into consideration.

However, to our best knowledge, there is no node-disjoint multipath receiver based forwarding solution suitable for such a comparison. Therefore, we have developed a node-disjoint solution based on VIRTUS, namely ND-VIRTUS. It focuses on choosing two paths with a maximum number of disjointed nodes. Similar to LIAITHON$^+$, each packet has a special Path ID that indicates the path it is supposed to pass along. Once a forwarder has reserved itself for forwarding packet with one Path ID, it maximizes its waiting time for scheduling packets with another Path ID. By this method, the forwarders used for one path are mostly not used for another path. The parameters of ND-VIRTUS are also shown in Table 4.4.

4.6.4.3 Performance Comparison

Figure 4.11, 4.12 and 4.13 have shown the performance comparison between LIAITHON$^+$ and VIRTUS, as well as ND-VIRTUS. As we can observe, LIAITHON$^+$ notably outperforms other solutions when data rate increases. The main reason for this enhancement is stated in the following. VIRTUS experiences severe collisions as well as congestions when the data rate reaches its load limit. The collisions are mainly between multiple forwarders. Although the problem of multiple forwarders is addressed in VIRTUS, it is inevitable in any receiver based forwarding solutions. Under heavy load conditions, collisions among multiple forwarders have significantly increased, which lead to even more forwarders and thus more collisions. LIAITHON$^+$ neatly distributes the heavy load into two almost decoupled paths, by this means solving the scalability problem of VIRTUS for high data rate conditions. However, as another multipath solution, ND-VIRTUS does not gain any performance improvement from load distribution. The problem for this solution is that the two selected paths are close or even twisted together regardless of the degree to which their nodes are disjointed, which causes serious route coupling effects.

Figure 4.11 shows the frame loss comparison among three works. VIRTUS has already achieved a great performance at 1000kbps or lower. As for LIAITHON$^+$, it does not only inherit this great performance at low data rates but also works well under higher data rates. However, we admit that the frame loss at very high data rates (i.e., 2000kbps or higher) is greatly increased. In the ideal case, the frame loss of 2-path LIAITHON$^+$ at 2000kbps should be as low as that of VIRTUS at 1000kbps. However, the collisions,

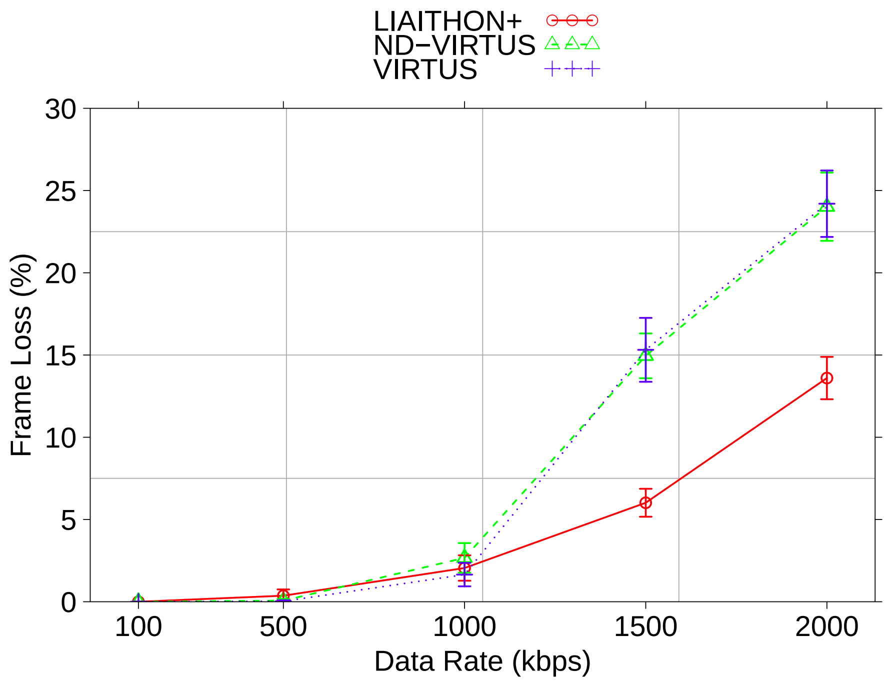


Figure 4.11: Performance evaluation: Frame Loss

mainly from route coupling between two paths, are incurred at a high data rate, especially in the area around the source node and the destination node.

As shown in Figure 4.12, when data rate is 500kbps or lower, the delay of LIAITHON$^+$ is higher than others. That is because, as we previously mentioned, the path length is increased when compared to VIRTUS. However, the delay in LIAITHON$^+$ is significantly smaller than the QoS requirement of video streaming (i.e., 4 to 5 seconds) defined by CISCO [19]. Under high data rate conditions, significant collisions happen around the shortest path area in VIRTUS. Thus, packets which have detoured are more likely to arrive at the destination; this causes VIRTUS to suffer from a higher delay at a higher data rate. However, the delay in LIAITHON$^+$ does not increase a lot at a high data rate, since the heavy traffic load is distributed as stated above.

As we can observe from Figure 4.13, LIAITHON$^+$ achieves a significant decrease in terms of cost when compared to others. The main reason is load distribution and the reduction of the route coupling effect as mentioned above. Another reason is that most

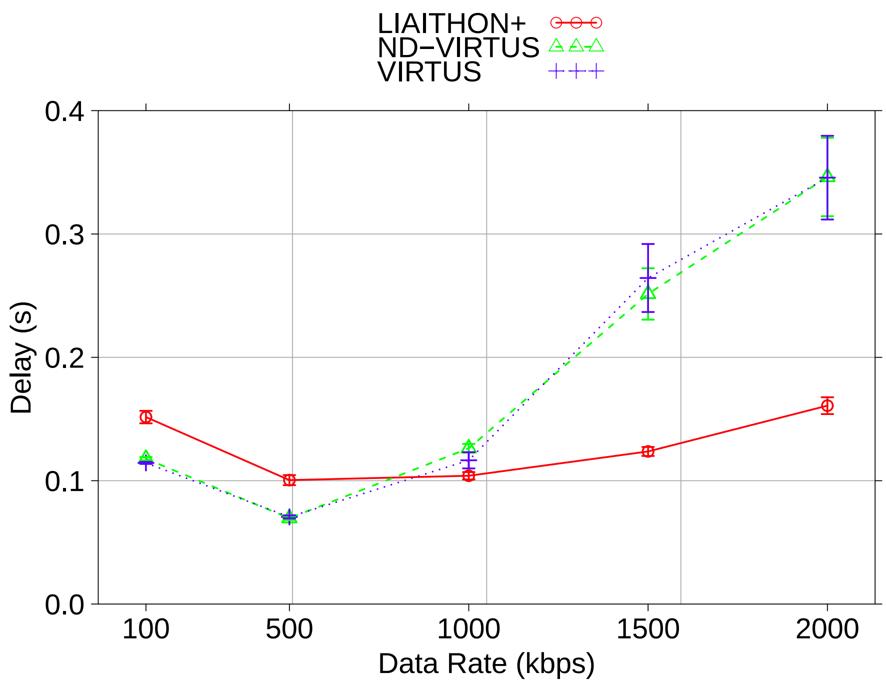


Figure 4.12: Performance evaluation: Delay

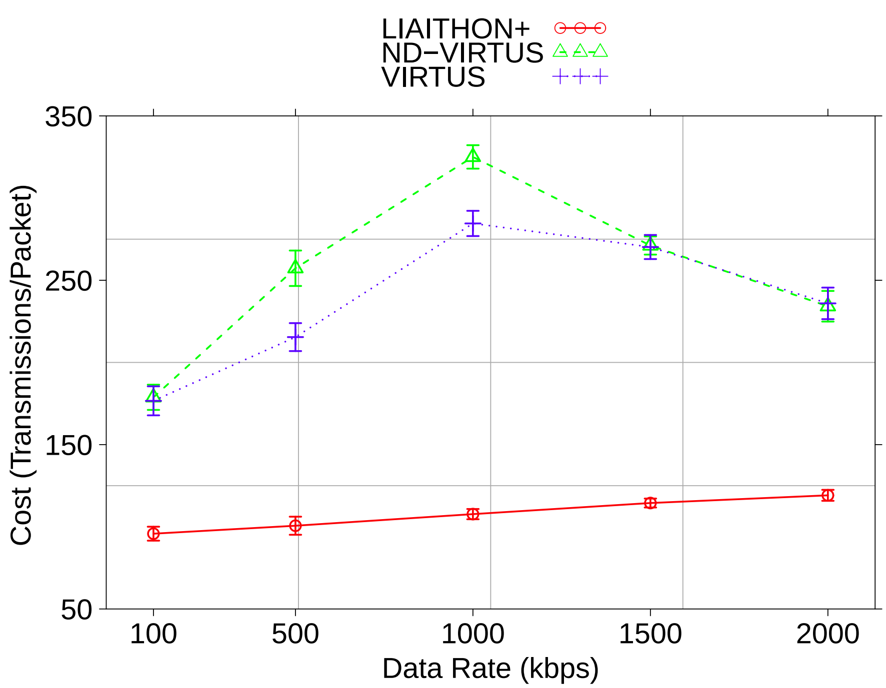


Figure 4.13: Performance evaluation: Cost

of the chosen forwarders are located around the optimal path, as we can observe in the multipath graph (Figure 4.7). This prevents the packets from unnecessarily traversing over the whole network area.

4.6.4.4 Summary

The performance evaluations indicate that the 2-path LIAITHON$^+$ significantly outperforms the underlying single path solution as well as the node-disjoint solution. From the simulation results, we can see that LIAITHON$^+$ is able to achieve video streaming task for urban vehicular ad hoc networks with a considerably small frame loss, delay and cost at 1500kbps or lower. One thing that can be improved upon in LIAITHON$^+$ is the performance under relatively high data rate conditions (i.e., 2000kps or even higher) in terms of frame loss.

4.7 Summary

This chapter describes the proposed multipath model, namely LIAITHON$^+$. Towards the objective of discovering an appropriate video streaming solution over urban VANETs, LIAITHON$^+$ is , to our best knowledge, the first work that incorporates the multipath concept into the receiver based forwarding scheme. This task is accomplished through the proposed metric, degree of closeness. Moreover, unlike most of the existing multipath solutions, LIAITHON$^+$ takes into consideration the route coupling effect and path length growth, in order to maximize the advantage of using multiple paths. Additionally, the metrics for greedy and link durability strategies, used in VIRTUS, are also leveraged in LIAITHON$^+$.

Based on this model, we have developed two solutions, 2-path and 3-path LIAITHON$^+$, in order to evaluate the impact of path number growth. Each of these solutions has been tailored with a location-aware waiting time calculation scheme and a route coupling prevention scheme. Apart from the normal waiting time scheme, the proposed area-aware waiting time schemes adopt different calculation methods according to different areas in which the intermediate nodes lies. These schemes could, to some extent, tackle the high collision rate problem around the source node and the destination node. In addition, the proposed route coupling prevention schemes further address the route coupling effect. The impact of the path number growth is studied through implementations and simulations. The results have shown that the appropriate number of paths is two. This is mainly due to the growth of path length and the increase of route coupling effect, when the path number goes up.

The performance evaluation has shown that the 2-path LIAITHON$^+$ notably outperforms its underlying single path work, VIRTUS. It also significantly outperforms the node-disjoint solution, ND-VIRTUS. Furthermore, it is able to accomplish the task of video streaming over urban VANETs at 1500kbps or lower data rate (delay: within 0.15 s; frame loss: within 6%; cost: within 120 transmissions/packet).

Chapter 5

The Impact of Added Redundancy on LIAITHON$^+$

In order to control frame loss, retransmission schemes are commonly used, such as in TCP (Transmission Control Protocol). By sending acknowledgments to the source node, a destination node can inform the source node either of the lost packets or of the successfully received packets. The source then retransmits the lost packets to the destination in order to recover from the frame loss. However, this scheme is rarely used in video streaming. The acknowledgment and retransmission phases are extremely time consuming; this is definitely not suitable for delay sensitive video streaming. Additionally, the dynamic topology of VANETs makes this scheme extremely unreliable.

In contrast, adding redundancy controls frame loss without incurring excessive delay. Therefore, it is more preferred in video streaming solutions. The source node will send redundant data to the destination along with the original video data. The destination is then able to recover all the video data from part of the video data and the redundant data it received.

Based on 2-path and 3-path LIAITHON$^+$ presented in Chapter 4, this chapter further exploits the impact of added redundancy on the proposed multipath scheme. Section 5.1 describes the methods used to adopt redundancy in LIAITHON$^+$. Section 5.2 shows the simulation results and discusses the impact of added redundancy on LIAITHON$^+$.

5.1 Redundancy Method Description

To adopt the redundancy method in LIAITHON$^+$, three major design issues are confronted, which are addressed in the following questions. What is the redundant packet composed of? How are redundant packets sent along with the original packets? How should received packets be handled at the destination?

5.1.1 Recovery Packet

A simple way to add redundancy is to send copies of the original packets. It is better to apply this method when the redundancy ratio, i.e., the ratio of the number of redundant packets to the number of original packets, is multiple of 100%. This is because, only in this way, the redundancy is able to cover all the original data uniformly. However, a large redundancy ratio is not suitable for LIAITHON$^+$ which has already achieved a low frame loss.

As a result, a small redundancy ratio with uniform coverage is demanded in LIAITHON$^+$. To achieve this task, we use erasure code [58], in which the original data is transformed into a larger data with redundancy. In this case, the original data can be recovered from the subset of the transformed data. The redundancy packet, named recovery packet in our approach, has the same packet header (excluding packet type) as the original video packet; though it does have a different payload. The payload (p_r) of the recovery packet is the XOR-sum (sum of exclusive or operation) of the payloads ($p_j ... p_{j+n}$) which are from a set of sequential original video packets:

$$p_r = p_j \oplus p_{j+1} \oplus ... \oplus p_{j+n} \tag{5.1}$$

We name this series of original packets and their recovery packet a block. The size of the block is the number (n) of the original packets it contains . According to the property of XOR operation, any video payload can be recovered as long as the other packets in the same block are received:

$$p_i = p_r \oplus p_j \oplus ... \oplus p_{i-1} \oplus p_{i+1} \oplus ... \oplus p_{j+n}\ ;\ j \leq i \leq j+n \tag{5.2}$$

5.1.2 Packets Sending

We assigned the total video packets into a set of blocks with the same size (excluding the last block). Every block has been allocated a recovery packet, which is responsible for recovering packet loss in its block[1]. The recovery packet is the last packet in its block and thus it is transmitted after all the other packets. Since every block is the same size, the block size is the only parameter determining the redundancy ratio. A smaller block size will lead to a higher redundancy ratio, and vice versa.

There are two reasonable ways to add redundancy into LIAITHON$^+$. The first solution, named 2-path LIAITHON$^+$ with redundancy (2LR), is based on 2-path LIAITHON$^+$. The video source simply treats the recovery paths as normal video packets and transmits all the packets on the two chosen paths in turns. The same concept can be used on 3-path LIAITHON$^+$ as well. However, this solution should acquire lower performance compared to 2LR, which is deduced from the comparison between 3-path and 2-path LIAITHON$^+$ (Subsection 4.6.3). This mainly results from the increase of the route coupling effect. A simple way to alleviate the severe route coupling effect between the center path and the side paths is to reduce the number of packets transmitted on the center

[1] There is no recovery packet for the last block.

path. Since the redundancy ratio demanded in LIAITHON$^+$ is low, we can use the center path to transmit the recovery packets only and use the side paths to transmit the original video packets. This carries out the second proposed solution, which is called 3-path LIAITHON$^+$ with the center path for redundancy (3LCR).

The introduction of redundancy leads to a problem, which is more packets are sent at the source. This decreases the data rate of the video packets' transmission, if the source does not change its packets' sending rate. In other words, the data rate for the source node to send packets, we call it the overall data rate, is used for both the video packets and the recovery packets. Therefore, when evaluating the benefit of added redundancy, it should be compared with the original work under the same video packets' data rate instead of the overall data rate. By inputing the expected video packets' data rate (DR_{video}) and the block size (n), the overall data rate ($DR_{overall}$) can be calculated as follows:

$$DR_{overall} = DR_{video} \times \frac{(PacNum \setminus n) \times (n+1) + PacNum\%n}{PacNum} \tag{5.3}$$

where $PacNum$ is the number of the original video packets. The source node uses the calculated overall data rate to transmit packets. From the equation, we can observe there exists a trade-off between increasing the redundancy ratio and decreasing the overall data rate. This trade-off can be adjusted through the block size. A smaller block size can offer more recovery packets, while it will increase the overall data rate, which leads to a higher collision rate. On the other hand, a larger block size leads to less redundancy as well as a lower collision rate. Thus, the performance study of the impact on the block size is a demanding topic.

5.1.3 Packets Reception

Once the destination node receives a packet, it will follow the flow chart depicted in Figure 5.1 to process the packet. Packets which may be used for recovery are temporarily

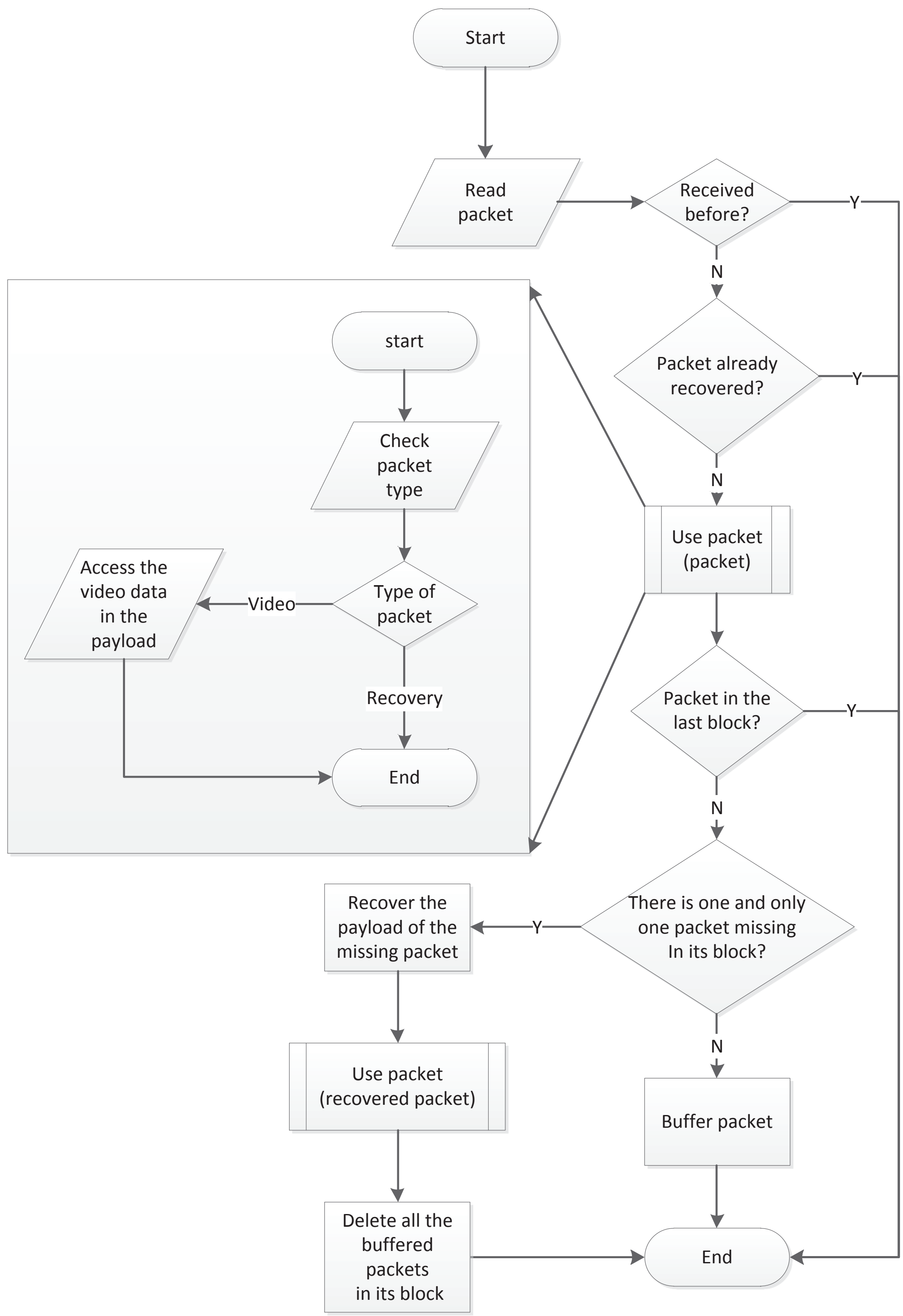


Figure 5.1: Flow chart of packets reception

buffered at the destination node. The payload of the missing packet will be recovered and used as soon as all other packets in its block are buffered. After the recovery process is finished, the buffered block of packets will be deleted.

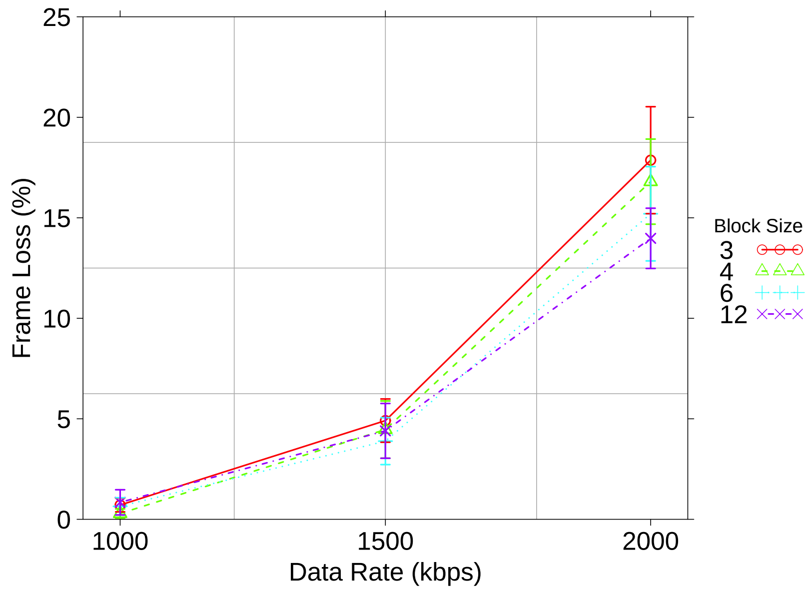
5.2 Simulations and Results

In order to evaluate the impact of added redundancy, we have implemented and simulated both 2LR (2-path LIAITHON$^+$ with redundancy) and 3LCR (3-path LIAITHON$^+$ with the center path for redundancy). Since the 2LR and 3LCR are based on LIAITHON$^+$, the same parameters, which are shown in Table 4.3, are applied to them. Also, the simulations are under the same simulation environment set in Subsection 4.6.1.

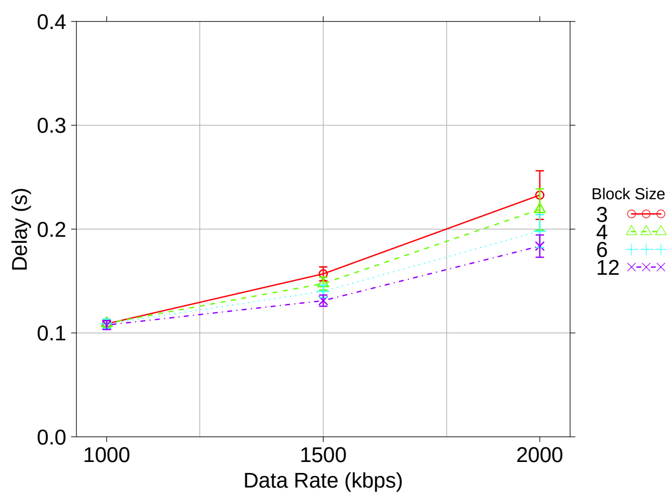
In this section, we first exploit the influence of block size on both solutions. After that, we compare the original LIAITHON$^+$ with 2LR and 3LCR, each under appropriate block sizes. It is worth mentioning, as explained before, all these solutions are compared under video packets' data rate, which we simply refer to as data rate in the following subsections. The data rate scales from 1000kbps to 2000kbps, since LIAITHON$^+$ already achieves very considerable performance at a lower data rate.

5.2.1 Impact of Block Size

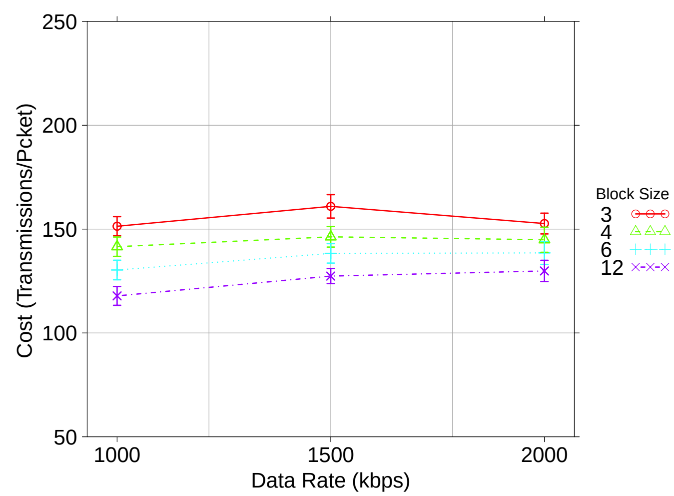
As discussed previously, the block size determines the redundancy ratio as well as the overall data rate. The fewer the packets that are contained in one block, the higher the redundancy ratio and overall data rate become. Thus, purely increasing the redundancy ratio, through reducing the block size, may not lead to higher performance. As a result, we investigate each redundancy solution under different block sizes. The redundancy ratio for a solution with block size n approximately equals $\frac{1}{n}$. Based on this approximation, the block sizes selected for comparison are 3, 4, 6 and 12, since the redundancy ratios for them are evenly spaced with an equal interval: $\frac{1}{12}$.



(a) Frame Loss

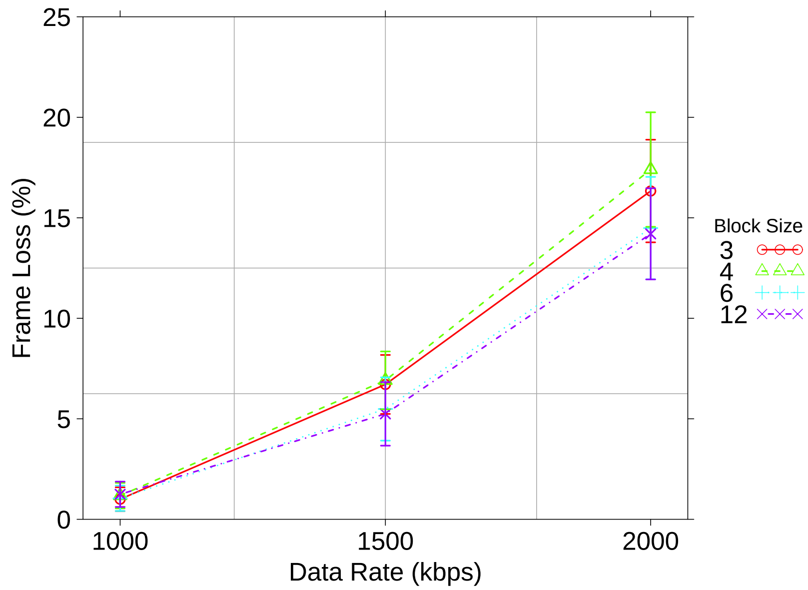


(b) Delay

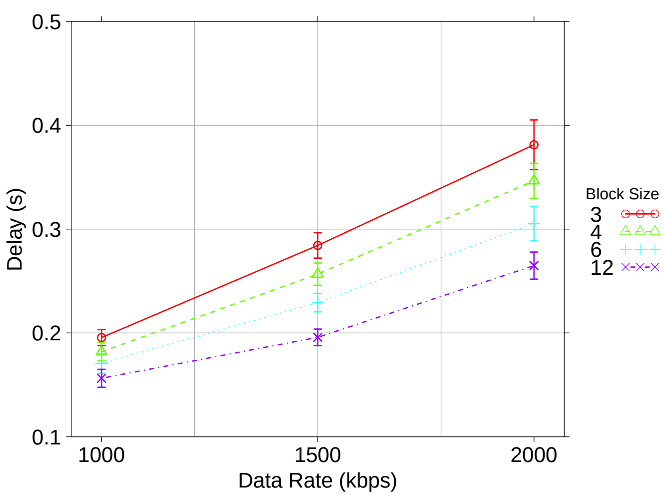


(c) Cost

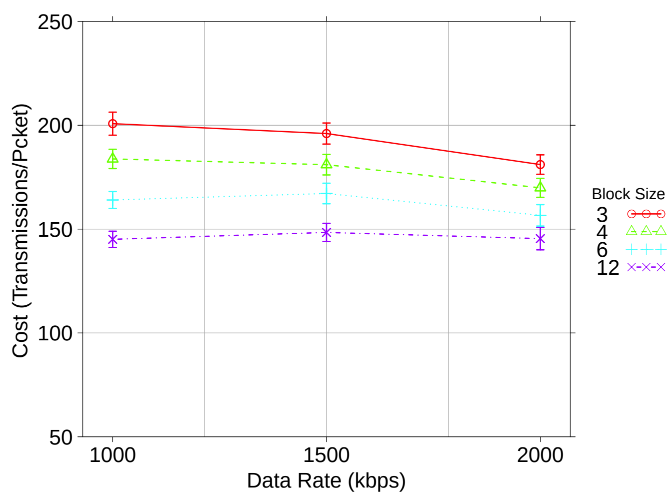
Figure 5.2: 2LR with different block sizes



(a) Frame Loss



(b) Delay



(c) Cost

Figure 5.3: 3LCR with different block sizes

Figure 5.2 and 5.3 have shown the impact of block size on 2LR as well as on 3LCR. 15 independent simulations, each under a different instance of the same mobility model, are conducted to plot every result. By comparing both solutions, we can see that the impact of block size on them is very similar. Thus, the following discussions apply for both solutions.

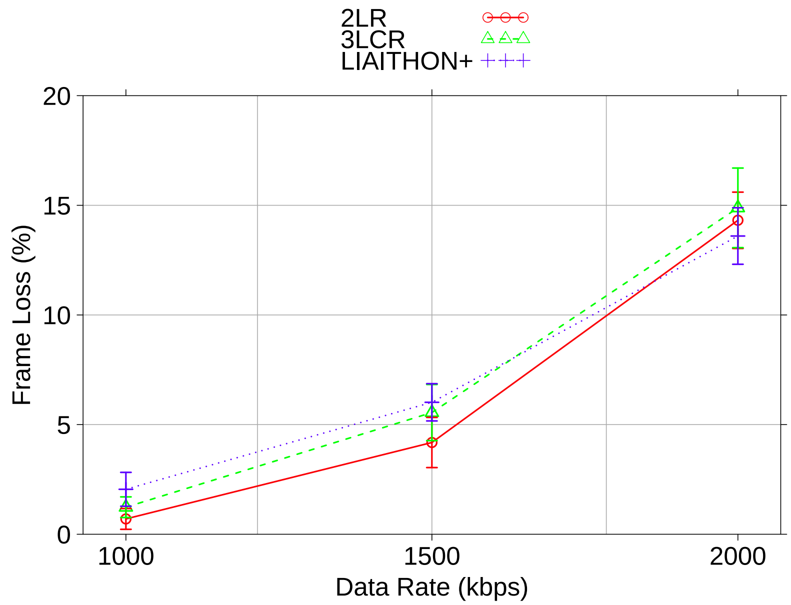
A smaller block size tends to acquire slightly lower frame loss at 1000kbps, while a larger block size acquires lower frame loss at higher data rates. The major reason is stated as follows. At a low data rate, the disadvantage of a smaller block size, which is the increase of the overall data rate, is eclipsed by its advantage, which is the increase of the redundancy ratio. While at a high data rate condition, LIAITHON$^+$ already faces its bottleneck, due to route coupling. Thus, the increase in the overall data rate significantly exacerbates this problem.

Another observation is that the smaller the block size is, the higher the delay and cost will be. This higher delay for smaller block size is mainly due to the higher collision rate caused by the higher overall data rate. As for the cost, decreasing the block size not only increases the collision rate but also results in more packets being transmitted.

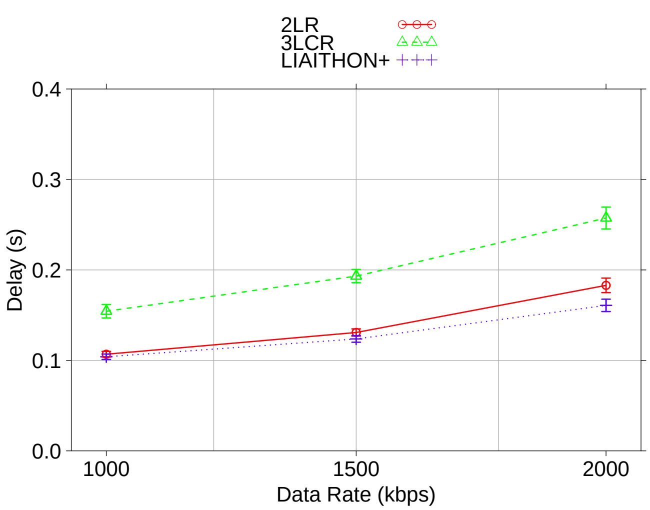
We selected 12 as the appropriate block size discovered for both 2LR and 3LCR solutions. Although the solutions with block size of 12 are slightly inferior in terms of frame loss at lower data rates, they have achieved much lower frame loss at higher data rates. Besides, they have the crucial advantages at delay and cost.

5.2.2 Performance Evaluation

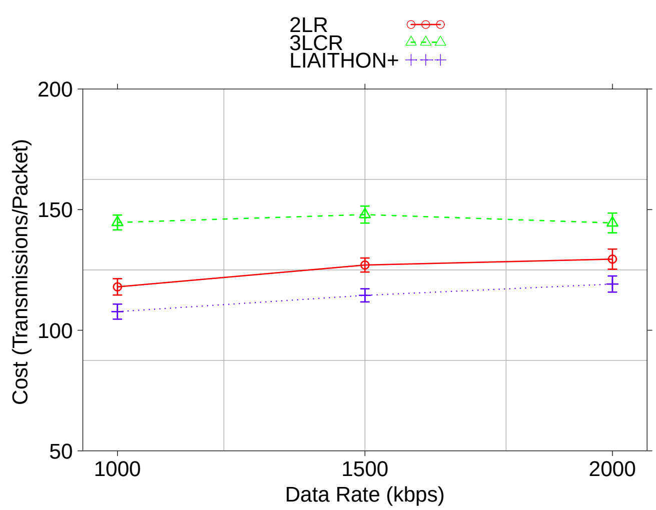
In order to evaluate the performance impact of added redundancy, we compare the redundancy solutions to the original LIAITHON$^+$. To be specific, the three works to be compared respectively are, 2LR with a block size of 12, 3LCR with a block size of 12 and 2-path LIAITHON$^+$. All plotted results are averages of 20 independent results, each under a different instance of the same mobility model. Figure 5.4 depicts the performance



(a) Frame Loss



(b) Delay



(c) Cost

Figure 5.4: Impact of added redundancy

comparison in terms of frame loss, delay and cost.

According to Figure5.4 (a), added redundancy achieves a performance gain at 1000kbps and 1500kbps in terms of frame loss. The improvement is not as great as we expected, since there is very limited room for increase due to the considerable performance of 2-path LIAITHON^{+}. At 2000kbps, the added redundancy has a negative impact on the frame loss. The main reason is similar to the one that accounts for the increase of frame loss when block size decreases at 2000kbps. The gain of added redundancy is exceeded by the increase of the collision rate which is caused by higher overall data rate. The overall data rate for 2-path LIAITHON^{+} is the same as the video data rate which is 2000kbps. However, the overall data rate for both 2LR and 3LCR calculated from Equation 5.3 is 2164kbps. LIAITHON^{+} already faces severe collisions, especially from route coupling between multiple paths, at 2000kbps; thus, a 164kbps increase of data rate significantly aggravates this problem. To put it simply, the balance between providing recovery packets and incurring a higher collision rate biases the former one at 1500kbps or lower while favoring the latter one when the data rate goes higher.

By adding redundancy, the delay and the cost are increased. This complies with the explanations which are more packets are sent as well as more collisions are incurred. However, the increase introduced is within an acceptable range, especially that of 2LR.

Another observation is that 2LR outperforms 3LCR. This indicates that, even with less packets sent on the center path, the 3LCR still faces the same problems with the 3-path LIAITHON^{+}. These problems mainly are the increase of path length as well as that of the route coupling effect described in Subsection 4.6.3. According to the performance study, we suggest the use of added redundancy on 2-path LIAITHON^{+} (i.e., 2LR) at a data rate of 1500kbps or lower.

5.3 Summary

This chapter studies the impact of added redundancy on LIAITHON$^+$. Two proposed redundancy solutions, 2LR and 3LCR are presented in detail. They both use the XOR based erasure code to generate the recovery packets. As for the packets' sending scheme, 2LR uses the pair of paths discovered in 2-path LIAITHON$^+$ to send both the original video packets and the recovery packets. While based on 3-path LIAITHON$^+$, 3LCR uses the center path for recovery packets and the side paths for video packets.

Through simulations, we have investigated the impact of the block size, which determines both redundancy ratio and the overall data rate, on both solutions. After that, the impact of added redundancy is studied by comparing 2LR and 3LCR with the original LIAITHON$^+$. The results have shown that the added redundancy has a positive impact when the data rate increases up to around 1500kbps, while it has a negative impact at a higher data rate. Although the improvement of the added redundancy is not impressive, we have provided deep insight into this issue.

Chapter 6

Conclusion and Future work

Due to the diverse challenges facing vehicular networks and the strict QoS requirements of video streaming, existing works are not competent enough for the task of video streaming over urban VANETs. In this thesis, a new multipath solution, LIAITHON$^+$, is proposed for the purposed of filling this gap. Moreover, LIAITHON$^+$ is further developed through the performance study on the impact of added redundancy. This chapter gives a summary of the thesis, followed by potential future directions for this research.

6.1 Conclusion

In this thesis, the background information of VANETs has been introduced in Chapter 1. After that, a literature review is shown in Chapter 2, in terms of the current research status about video streaming over VANETs as well as multipath transmission solutions. Enabling video streaming over vehicles is intended for the purpose of providing fundamental support to a variety of applications in VANETs. However, the VANETs environment presented various challenges, such as its highly dynamic topology and its non-uniform distribution. Moreover, the stringent QoS of video streaming, in terms of frame loss and delay, significantly hindered many potential solutions. To address these challenges, the greedy forwarding strategies as well as receiver based forwarding schemes are preferred. Additionally, the consideration of link durability is also required in support of the continuous video content's transmission. The underlying single path work, VIRTUS, described in Chapter 3, takes into consideration all of these issues. Besides, the major problems of receiver based solutions are tackled in VIRTUS as well. The results have shown that it indeed achieves remarkable performance, in terms of frame loss, delay and cost, at data rate of 1000kbps or lower. However, when the data rate goes higher, its performance faces a significant drop due to network collisions.

With the demand for providing a more reliable solution for video streaming over VANETs, a brand new multipath solution, LIAITHON$^+$, is proposed in Chapter 4. By distributing heavy traffic load into multiple paths, the collision problem in the single path solution can be largely alleviated. However, to the best of our knowledge, there is no existing multipath solution that relies upon the receiver based scheme. Moreover, through the literature review in Chapter 2, most of the works only use node-disjoint or link-disjoint metrics to discover the multiple paths. However, the route coupling effect, which has been proven as the major obstacle, is overlooked.

These issues are directly addressed in LIAITHON$^+$. Three metrics, geographic advance, link stability and degree of closeness, are considered when selecting the forwarding nodes. As the key metric of LIAITHON$^+$, degree of closeness is responsible for discovering a set of relatively short paths with a minimal route coupling effect. In order to discover the suitable path number, 2-path and 3-path solutions are proposed, each of them has a specifically tailored area-aware waiting time calculation scheme and a route coupling prevention scheme. These schemes ensure that the chosen paths are maximally decoupled, especially in the areas around the source node and the destination node. The performance is evaluated through the simulation under NS2. The results have shown that the appropriate path number is 2, because solution with a higher path number is prone to longer paths and have a higher degree of route coupling effect. The performance of 2-path LIAITHON$^+$ is compared with VIRTUS and the node-disjoint solution. LIAITHON$^+$ is observed with a considerable improvement over other solutions, especially for performances at high data rates. As indicated in the results, LIAITHON$^+$ can be regarded as a very promising solution for the task of video streaming over urban VANETs when data rate scales up to 1500kbps.

Motivated by further improvements, the impact of the added redundancy on LIAITHON$^+$ is studied in Chapter 5. Towards this objective, two approaches, 2LR based on 2-path LIAITHON$^+$ and 3LCR based on 3-path LIAITHON$^+$, are proposed. In 2LR, the redundancy, i.e., recovery packets, are regarded as the original packets and transmitted along the chosen paths. In 3LCR, the center path is used for the transmission of the recovery packets and the side paths are for the video packets. The simulation results have indicated that added redundancy gains improvement, in terms of frame loss, when data rate scales up to 1500kbps. While at higher data rates, added redundancy faces its limitation which is mainly caused by the increase of the collision rate.

6.2 Future Work

This thesis has achieved solid improvements towards the target of providing an appropriate multipath solution for video streaming over urban VANETs. Moreover, it opens up a wide variety of interesting directions for further exploration. The open issues are, but are not restricted to, the following:

Support for 3D stereo video streaming. With the development of high-end theaters and display devices, stereo video, which offers users the 3D depth perception, has stepped into people's life. Our solution could be extended to support 3D video streaming by transmitting right and left view stream on two chosen paths respectively.

The exploitation of multipath solution's use in the multichannel model. Some works (e.g., [59]) suggest the use of multichannel in order to avoid the co-channel interference between multiple paths, i.e., route coupling effect. However, different channels are still subject to inter-channel interference caused by multichannel overlapping. Authors in [60] have measured the interference level between adjacent wireless channels. They suggest the use of multichannel with separation in spatial or frequency domain. Thus, our work, LIAITHON$^+$, can be extended to support the multichannel model for the purpose of spatially separating the multiple channels.

Extending the use of multipath solution to support heterogeneous networks. In this thesis, we assume that all the vehicles have installed the same transmitter and receiver devices with the same transmission range. In the future, we are going to exploit the use of LIAITHON$^+$ for the heterogeneous network which consists of different radios as well as cognitive radios.

Index

Bibliography

[1] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: applications and related technical issues," *Communications Surveys Tutorials, IEEE*, vol. 10, no. 3, pp. 74 –88, quarter 2008.

[2] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE*, vol. 46, no. 6, pp. 164 –171, june 2008.

[3] J. Zhu and S. Roy, "Mac for dedicated short range communications in intelligent transport system," *Communications Magazine, IEEE*, vol. 41, no. 12, pp. 60 – 67, dec. 2003.

[4] IEEE P802.11p/D3.0, "Draft amendment to standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications-amendment 7: Wireless access in vehicular environment," 2007.

[5] R. Uzcategui and G. Acosta-Marum, "Wave: A tutorial," *Communications Magazine, IEEE*, vol. 47, no. 5, pp. 126 –133, may 2009.

[6] IEEE P1609.1, "Trial-use standard for wireless access in vehicular environments (wave) - resource manager," 2006.

[7] IEEE P1609.2, "Trial-use standard for wireless access in vehicular environments (wave) - security services for applications and management messages," 2006.

[8] IEEE Std P1609.3, "Ieee trial-use standard for wirelessaccess in vehicular environments (wave)-networking services," 2007.

[9] IEEE P1609.4, "Trial-use standard for wireless access in vehicular environments (wave) - multi-channel operation," 2006.

[10] CAR 2 CAR Communication Consortium, "http://www.car-to-car.org."

[11] Streaming Media, "http://en.wikipedia.org/wiki/video_streaming."

[12] M. Lindeberg, S. Kristiansen, T. Plagemann, and V. Goebel, "Challenges and techniques for video streaming over mobile ad hoc networks," *Multimedia Systems*, vol. 17, no. 1, pp. 51–82, 2010. [Online]. Available: http://www.springerlink.com/index/10.1007/s00530-010-0187-8

[13] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE*, vol. 2, no. 2, pp. 12 –22, june 2007.

[14] M. Roccetti, M. Gerla, C. Palazzi, S. Ferretti, and G. Pau, "First responders' crystal ball: How to scry the emergency from a remote vehicle," in *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa*, april 2007, pp. 556 –561.

[15] C. Rezende, H. S. Ramos, R. W. Pazzi, A. Boukerche, A. C. Frery, and A. A. Loureiro, "Virtus: A resilient location-aware video unicast scheme for vehicular networks," in *IEEE International Conference on Communications*, 2012.

[16] H. Menouar, M. Lenardi, and F. Filali, "Improving proactive routing in vanets with the mopr movement prediction framework," in *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, june 2007, pp. 1 –6.

[17] J. Li and C. Chigan, "Achieving robust message dissemination in vanet: Challenges and solutions," in *IEEE Wireless Communication Magazine*, 2006.

[18] H. Wu, R. Fujimoto, and G. Riley, "Analytical models for information propagation in vehicle-to-vehicle networks," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 6, sept. 2004, pp. 4548 – 4552 Vol. 6.

[19] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology).* Cisco Press, 2004.

[20] C. Tee and A. Lee, "Survey of position based routing for inter vehicle communication system," in *Distributed Framework and Applications, 2008. DFmA 2008. First International Conference on*, oct. 2008, pp. 174 –182.

[21] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254. [Online]. Available: http://doi.acm.org.proxy.bib.uottawa.ca/10.1145/345910.345953

[22] A. Wahid, H. Yoo, and D. Kim, "Unicast geographic routing protocols for inter-vehicle communications: a survey," in *Proceedings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, ser. PM2HW2N '10. New York, NY, USA: ACM, 2010, pp. 17–24. [Online]. Available: http://doi.acm.org.proxy.bib.uottawa.ca/10.1145/1868612.1868616

[23] F. Xie, K. Hua, W. Wang, and Y. Ho, "Performance study of live video streaming over highway vehicular ad hoc networks," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, sept. 30 2007-oct. 3 2007, pp. 2121 –2125.

[24] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, feb 1999, pp. 90 –100.

[25] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing.* Kluwer Academic Publishers, 1996, pp. 153–181.

[26] M. Guo, M. Ammar, and E. Zegura, "V3: a vehicle-to-vehicle live video streaming architecture," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, march 2005, pp. 171 – 180.

[27] L. Briesemeister, L. Schafers, and G. Hommel, "Disseminating messages among highly mobile hosts based on inter-vehicle communication," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 522 –527.

[28] H. Menouar, M. Lenardi, and F. Filali, "Movement prediction-based routing (mopr) concept for position-based routing in vehicular networks," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, 30 2007-oct. 3 2007, pp. 2101 –2105.

[29] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Information Fusion, 2008 11th International Conference on*, 30 2008-july 3 2008, pp. 1 –6.

[30] A. Capone, L. Pizziniaco, I. Filippini, and M. de la Fuente, "A sift: an efficient method for trajectory based forwarding," in *Wireless Communication Systems, 2005. 2nd International Symposium on*, sept. 2005, pp. 135 – 139.

[31] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi, "Video transport over ad hoc networks: multistream coding with multipath transport," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 10, pp. 1721 – 1737, dec. 2003.

[32] A. Razzaq and A. Mehaoua, "Video transport over vanets: Multi-stream coding with multi-path and network coding," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, oct. 2010, pp. 32 –39.

[33] J. Tsai and T. Moors, "A review of multipath routing protocols : From wireless ad hoc to mesh networks," *Technology*, vol. 8, pp. 30–37, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.5817&rep=rep1&type=pdf

[34] M. Tarique, K. E. Tepe, S. Adibi, and S. Erfani, "Survey of multipath routing protocols for mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 32, no. 6, pp. 1125 – 1143, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804509001027

[35] X. Li and L. Cuthbert, "Stable node-disjoint multipath routing with low overhead in mobile ad hoc networks," in *Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings. The IEEE Computer Society's 12th Annual International Symposium on*, oct. 2004, pp. 184 – 191.

[36] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 10, 2001, pp. 3201 –3205 vol.10.

[37] W.-I. Lee, S. Chowdhury, G.-Y. Kee, W.-S. Baek, and J.-Y. Pyun, "Velocity aware multipath distance vector routing protocol for high mobility over vehicular ad-hoc networks," in *Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on*, june 2011, pp. 189 –194.

[38] X. Li and L. Cuthbert, "On-demand node-disjoint multipath routing in wireless ad hoc networks," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, nov. 2004, pp. 419 – 420.

[39] M. Pearlman, Z. Haas, P. Sholander, and S. Tabrizi, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, 2000, pp. 3 –10.

[40] X. Huang and Y. Fang, "Performance study of node-disjoint multipath routing in vehicular ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 4, pp. 1942 –1950, may 2009.

[41] S. Waharte and R. Boutaba, "Totally disjoint multipath routing in multihop wireless networks," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 12, june 2006, pp. 5576 –5581.

[42] Y.-H. Wang, H.-Z. Lin, and S.-M. Chang, "Interference on multipath qos routing for ad hoc wireless network," in *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, march 2004, pp. 104 – 109.

[43] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Network Protocols, 2001. Ninth International Conference on*, nov. 2001, pp. 14 – 23.

[44] R. Leung, J. Liu, E. Poon, A.-L. Chan, and B. Li, “Mp-dsr: a qos-aware multi-path dynamic source routing protocol for wireless ad-hoc networks,” in *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, 2001, pp. 132 –141.

[45] A. Nasipuri and S. Das, “On-demand multipath routing for mobile ad hoc networks,” in *Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, 1999, pp. 64 –70.

[46] Y. Chen, Z. Xiang, W. Jian, and W. Jiang, “An improved aomdv routing protocol for v2v communication,” in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 1115 –1120.

[47] D. Saha, S. Toy, S. Bandyopadhyay, T. Ueda, and S. Tanaka, “An adaptive framework for multipath routing via maximally zone-disjoint shortest paths in ad hoc wireless networks with directional antenna,” in *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*, vol. 1, dec. 2003, pp. 226 – 230 Vol.1.

[48] J. Galvez, P. Ruiz, and A. Skarmeta, “Spatially disjoint multipath routing protocol without location information,” in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, oct. 2008, pp. 570 –571.

[49] S. Bandyopadhyay, K. Hasuike, S. Horisawa, and S. Tawara, “An adaptive mac protocol for wireless ad hoc community network (wacnet) using electronically steerable passive array radiator antenna,” in *Global Telecommunications Conference, 2001. GLOBECOM ’01. IEEE*, vol. 5, 2001, pp. 2896 –2900 vol.5.

[50] T. Ueda, K. Masayama, S. Horisawa, M. Kosuga, and K. Hasuike, “Evaluating the performance of wireless ad hoc network testbed with smart antenna,” in *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, 2002, pp. 135 – 139.

[51] D. Simon, “Optimal state estimation: Kalman, h infinity, and nonlinear approaches,” in *Wiley-Interscience*, 2006.

[52] Network Simulator (ns-2), “http://www.isi.edu/nsnam/ns/.”

[53] C. Rezende, A. Boukerche, R. W. Pazzi, B. P. Rocha, and A. A. Loureiro, “The impact of mobility on mobile ad hoc networks through the perspective of complex networks,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 9, pp. 1189 – 1200, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731510002789

[54] J. Klaue, B. Rathke, and A. Wolisz, “Evalvid - a framework for video transmission and quality evaluation,” in *In Proc. of the 13th International Conference on*

Modelling Techniques and Tools for Computer Performance Evaluation, 2003, pp. 255–272.

[55] R Development Core Team, "R: A language and environment for statistical computing." Vienna, Austria: R Foundation for Statistical Computing, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[56] Student's t-distribution, "http://en.wikipedia.org/wiki/student's_t-distribution."

[57] T. Williams, C. Kelley, and many others, "Gnuplot 4.4: an interactive plotting program," March 2010.

[58] Erasure Code, "http://en.wikipedia.org/wiki/erasure_code."

[59] B. Yan and H. Gharavi, "Multi-path multi-channel routing protocol," in *Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications*, ser. NCA '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 27–31. [Online]. Available: http://dx.doi.org/10.1109/NCA.2006.41

[60] O. Incel, S. Dulman, P. Jansen, and S. Mullender, "Multi-channel interference measurements for wireless sensor networks," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, nov. 2006, pp. 694 –701.