國 立 交 通 大 學

資訊科學與工程研究所

博 士 論 文

適用於都會區車載網路的

動態覆蓋式群播機制

與基於道路之預測式多路徑繞徑協定

Dynamic Overlay Multicast and

Road-based Predictive Multipath Routing

in Urban VANETs

研 究 生：謝宜玲

指導教授：王國禎　博士

中 華 民 國 一 〇 二 年 一 月

適用於都會區車載網路的
動態覆蓋式群播機制
與基於道路之預測式多路徑繞徑協定

Dynamic Overlay Multicast and Road-based Predictive Multipath Routing in Urban VANETs

研 究 生：謝宜玲　　Student: Yi-Ling Hsieh

指導教授：王國禎　　Advisor: Kuochen Wang

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

博 士 論 文

A Dissertation

Submitted to Institutes of Computer Science and Engineering

Department of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

January 2013

Hsinchu, Taiwan, Republic of China

中華民國一〇二年一月

適用於都會區車載網路的
動態覆蓋式群播機制
與基於道路之預測式多路徑繞徑協定

學生：謝宜玲　　指導教授：王國禎 博士

國立交通大學資訊科學與工程研究所

摘　要

在車載網路中，多媒體串流及群組通訊為兩種重要且有潛力的服務，而一個高效能的繞徑協定將能顯著地提升來源端至接收端的多媒體串流效能。因此，在本論文中，在應用層我們提出一個動態覆蓋式群播機制來提供實況多媒體串流給群組節點；而在網路層我們提出一個新穎的服務品質感知之基於道路多路徑繞徑協定。我們所提出的這兩個方法都旨在都會區車載網路中提供流暢的多媒體串流服務。

在都會區車載網路的各種應用服務中，提供資訊娛樂服務(infotainment service)已經是可預見的趨勢，而多媒體串流是潛力極高的資訊娛樂服務。我們考慮的情境是在群組車輛間進行實況多媒體串流群播服務，為此我們使用動態應用層覆蓋(dynamic application-layer overlay)方式來達成。由於非群組中的車輛不一定願意合作來協助群播，因此以應用層覆蓋式群播的方式會比其他種類的群播方式，如基於網路編碼之群播以及網路層群播，更具彈性。因此，為了適應都會區車載網路高移動性及障礙物密集的特性，我們提出了一個稱作 OMV (Overlay Multicast in VANETs)的高效能動態覆

蓋式群播機制來提供多媒體串流。在這個 OMV 方法中，我們提出兩項策略來增強覆蓋(overlay)的穩定性：(1)滿足服務品質之動態覆蓋及(2)網狀結構之覆蓋。滿足服務品質之動態覆蓋的策略是按照封包遺失率及端到端延遲時間來挑選潛在的新父節點，以優化此覆蓋架構。而網狀結構之覆蓋的策略是允許子節點可擁有多個父節點。我們採用了兩個真實的影音短片來評估及驗證 OMV 在多障礙物的都會區車載網路中傳送影音的可行性。評估結果顯示，相較於目前在車載網路上最佳的方法—Qadri 等人使用靜態網狀覆蓋的研究，平均來說 OMV 降低了 27.1%的封包遺失率、11.7%的端到端延遲時間，但只增加 2.1%的額外控制負荷。我們並將樹狀 OMV 與 ALMA 作比較，其中 ALMA 為另一個樹狀覆蓋式群播且是行動隨意網路(MANETs)中的最佳方法，OMV 降低了 7.1%的封包遺失率及 13.1%的端到端延遲時間。此外，針對都會區車載網路中障礙物密集的問題，我們歸納出在不同的道路疏密度下可行之串流速度與群組大小。據我們所知，現有的車載網路文獻中尚未有如何動態地調整應用層群播覆蓋架構，以有效傳送多媒體串流的研究。總結來說，為處理都會區車載網路中高變動的拓樸，我們提出一個能滿足服務品質的策略，使群組節點可切換至能提供較佳服務品質的新父節點。我們所提出的 OMV 適用於實況多媒體串流，例如緊急實況影音傳輸及同一群組車輛的遊客們實況影音導遊。

至於在網路層中，穩定且效能高的繞徑是車載網路成功的關鍵。現有文獻已經證明基於道路的繞徑方式是非常適用於車載網路。此外，一旦使用中的路徑失敗時，多路徑繞徑方式可提供額外的路徑供切換。然而，現有的多路徑繞徑協定皆為基於節點的繞徑方法，因此並不適用於都會區之車載網路。在本論文中，我們也提出了一個適用於都會區車載網路之新穎的服務品質感知之基於道路多路徑繞徑協定(RMRV)。我們所提出的 RMRV 能找到多條路徑並且被有效地使用。我們提出了一個新穎的時空平面圖

(space-time planar graph)的方法來預測一條路徑中各個路段的連通性，以得到此路徑未來的各個存活時段(life periods)。如此一來，源節點(source node)可動態地切換使用一條目前是連通且延遲時間較短的路徑，我們稱此為基於服務品質之路徑切換方法(QoS path switching)。採用此方法可預先避免封包遺失及可選用潛在延遲時間較短的路徑。模擬結果顯示，相較於另一個代表性的基於道路單路徑之繞徑協定，RBVT-R，我們所提出之 RMRV（一次只使用一條路徑來傳送資料）提高了 16.6%的封包送達率、降低了 35.3%的端到端延遲時間，以及減少了 45.8%的額外控制負荷。據我們所知，現有文獻中尚未有適用於車載網路的基於道路之多路徑繞徑協定。

關鍵詞：多路徑，服務品質，基於道路，繞徑，動態覆蓋式群播，多媒體串流，都會區車載網路。

Dynamic Overlay Multicast and Road-based Predictive Multipath Routing in Urban VANETs

Student: Yi-Ling Hsieh Advisor: Dr. Kuochen Wang

Department of Computer Science and Engineering
National Chiao Tung University

Abstract

In VANETs (Vehicular Ad Hoc Networks), multimedia streaming and group communication are two important potential services. An efficient routing protocol can significantly enhance source-destination multimedia streaming performance. Therefore, in the application layer we propose a dynamic overlay multicast scheme for live multimedia streaming among a group of nodes, and in the network layer we propose a novel road-based QoS-aware multipath routing protocol. Both the proposed approaches aim to provide smooth multimedia streaming in urban VANETs.

For various application layer services, infotainment service has been a foreseeing trend in VANETs, and multimedia streaming has a high potential in VANET infotainment service. We consider the scenario of live multimedia streaming multicast to vehicles of the same group using a dynamic application-layer overlay. Due to the willingness for cooperation of non-group nodes, application-layer overlay multicast is more feasible than other kinds of multicast such as network-coding-based multicast and network-layer multicast. To adapt to high mobility and full of obstacles in urban VANETs, we propose an effective dynamic overlay multicast scheme for multimedia streaming, called OMV (*Overlay Multicast in*

VANETs). The proposed OMV enhances an overlay's stability with two strategies: (1) *QoS-satisfied dynamic overlay* and (2) *mesh-structure overlay*. The QoS-satisfied strategy to adjust the overlay selects potential new parents based on their streams' packet loss rates and end-to-end delays. The mesh-structure strategy allows a child to have multiple parents. We evaluate the proposed OMV in urban VANETs with obstacles using two real video clips to demonstrate the feasibility of the OMV for real videos. Evaluation results show that comparing the proposed OMV to Qadri *et al.*'s work, which is a static mesh overlay and is the best method available in VANETs, the packet loss rate is reduced by 27.1% and the end-to-end delay is decreased by 11.7%, with a small control overhead of 2.1%, on average. Comparing the proposed OMV for tree overlays to ALMA, which is for dynamic tree multicast overlays and is also the best method available in MANETs, the packet loss rate is reduced by 7.1% and the end-to-end delay is decreased by 13.1%. In addition, to address the problem of obstacle-prone urban VANETs, we also derive feasible stream rates and overlay sizes for city maps with different road section sizes. To the best of our knowledge, how to organize and dynamically adjust an application layer multicast overlay for live multimedia streaming have not been studied in existing VANET literatures. In summary, to deal with highly dynamic topologies in urban VANETs, we propose a QoS-satisfied strategy for group nodes to switch to new parents that can offer better QoS. The proposed OMV is feasible for live multimedia streaming applications, such as emergency live video transmission and live video tour guides for passengers in different vehicles that belong to the same multicast group.

As to the network layer, stable and efficient routing plays a key role for the success of VANETs. Road-based routing has been shown well-suited for urban VANETs, and multipath routing provides alternative routes once the current route fails. However, existing multipath routing protocols are node-based, which are not suitable for urban VANETs. In this dissertation, we propose a novel *road-based QoS-aware multipath routing protocol for urban VANETs* (RMRV). The proposed RMRV can find multiple paths and intelligently utilize them.

We also propose a s*pace-time planar graph* approach to predict the connectivity of each road section (RS) in a path, and then derive the path's future life periods. In this way, the source node may dynamically switch to use a path which is currently connected with shorter delay, which is called QoS path switching. QoS path switching avoids packet loss in advance and chooses the path with potential shorter delay. Simulation results show that comparing to a representative approach, RBVT-R, which is a single-path road-based routing protocol, the packet delivery ratio of the proposed RMRV (only one path used at a time) is 16.6% higher than that of RBVT-R, the average end-to-end delay of RMRV is 35.3% lower than that of RBVT-R, and the control overhead of RMRV is 45.8% lower than that of RBVT-R, on average. To the best of our knowledge, there is no existing road-based multipath routing protocol for VANETs.

Keywords: Multipath, QoS, road-based, routing, dynamic overlay multicast, multimedia streaming, urban VANETs.

誌　　謝

由衷感恩在博士班期間，所有幫助我的人事物，最終得以這本論文來總結及呈現這段期間的所學所成。能完成這份學業，首先感謝我的指導教授，王國禎老師的指導與照顧，總是以嚴謹的態度來幫助我完成每一篇論文，由衷謝謝您的指導與示範。而此生中有機會攻讀博士班，感恩我的靈性導師，　悟覺妙天師父，造化我、幫助我成就未來更高的成就，感恩　師父的栽培。而多年的求學過程裡，感恩家人給予我堅強的後盾，謝謝最愛我也是我最愛的媽媽與弟弟。此外並感謝學長黃貴笠、學弟莊宜達，常熱心回饋各種建議，以及模擬程式上的幫助。還有謝謝校內審查/口試委員，曾煜棋院長及趙禧綠教授，給予這份論文豐富的回饋，以及謝謝校外口試委員們，郭斯彥院長、許健平教授、林偉教授及楊竹星教授蒞臨驗證學生的博士論文研究成果，讓我有機會將這份研究成果充分的反芻檢視。還有許許多多給予我幫助的前輩、同儕，以及MBL實驗室的學長學弟妹，謝謝你們。最後，感謝國科會研究計畫(編號：NSC94-2213-E-009-043、NSC95-2221-E-009-022、NSC96-2628-E-009-140-MY3、NSC97-3114-E-009-001、NSC98-2219-E-009-008、NSC99-2219-E-009-006、NSC100-2219-E-009-004)長期的支持。

謹以此博士論文，獻給我最愛的家人，以及最敬愛的　師父。

Content

List of Figures

List of Tables

Chapter 1

Introduction

A VANET (Vehicular Ad Hoc Network) is a special class of MANETs (Mobile Ad Hoc Networks). VANETs are composed of vehicles so that through inter-vehicle communications, vehicles can exchange data with each other for purposes of safety or infotainment services. Nowadays, the feasibility of VANETs is promising due to deployment of ITS (Intelligent Transport System) infrastructure, availability of OBUs (on-board units) for vehicles, and emerging standards for inter-vehicle communications, such as IEEE 802.11p and IEEE 1609.1-4. Especially, group communication and multimedia streaming are important potential services for VANETs. Using application-layer overlay multicast to support multimedia streaming for a group of nodes in VANETs has been studied in literature and has shown its potential feasibility. Besides, in ad hoc networks, packets are node-by-node relayed from one node to another node; thus an efficient routing protocol can significantly enhance the source-destination data transmission performance. Therefore, aiming to provide smooth data transmission in urban VANETs, in network layer we propose a road-based QoS-aware multipath routing protocol, called RMRV, and in application layer we propose a dynamic overlay multicast scheme, called OMV, for live multimedia streaming among a group of nodes. As shown in Figure 1, we propose OMV for multimedia streaming service in application layer and in network layer we propose RMRV for packet routing. As to the other layers, we use standard protocols in this dissertation.

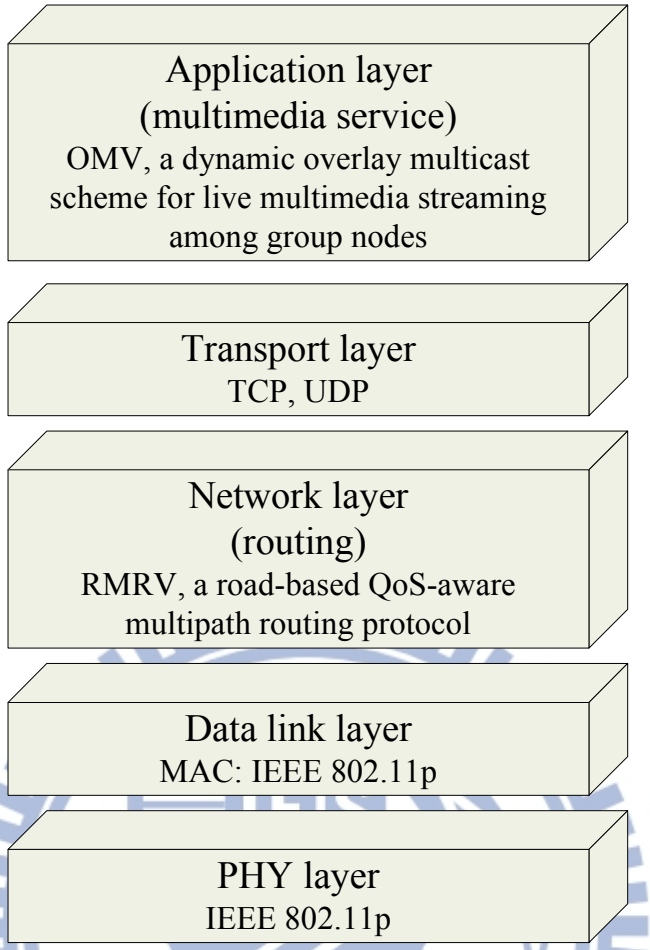


Figure 1. Illustration of the protocols proposed in different layers.

1.1 Overlay multicast in VANETs

Infotainment service is a foreseeing trend in VANETs. More and more studies reveal its feasibility and applicability [2][3][4][5][6][7][8][22]. The delivered infotainment data may be in a form of messages, files, or multimedia streaming. Multimedia streaming allows passengers not having to wait for video downloading to finish, and it enables live multimedia service.

1.1.1. Live multimedia streaming with overlay multicast in VANETs

This dissertation first considers the scenario of live multimedia streaming among vehicles of the same group using a dynamic overlay to support application layer multicast. With live multimedia streaming multicast, passengers in vehicles can join a tour guide's live video tour from a source vehicle. For example, in a bus trip that includes several buses, to save travel expenses, only one tour guide is hired to give live touring in one bus. The passengers in the rest of buses can view live touring video streaming from the bus with a tour guide. In this dissertation we focus on live streaming; however, if on-demand streaming is required, live streaming can be easily extended to support on-demand streaming by incorporating mechanisms such as retransmission and prefetch.

As shown in Figure 2, vehicles interested to join a multicast group can form an application-layer overlay. The overlay can be organized as a tree or mesh, and the video streaming source is the root of the overlay. The main advantage using an overlay is that it only requires group nodes to support the multicast overlay construction and maintenance, which eliminates the problem of intermediate *non-member* vehicles' willingness for cooperation. In VANETs, we cannot assume that all vehicles support a specific multicast protocol. By using an overlay that builds logical paths (i.e. overlay links) between group nodes, an advantage is that the overlay may still work well even if the group nodes are not dense enough. Another advantage of using an overlay is routing flexibility. Packet delivery among group nodes is handled by an underlying routing protocol that may cope with a dynamic VANET better. There has been various routing protocols designed for VANETs [12][13][14][15]. An overlay itself just needs to focus on application layer optimization.

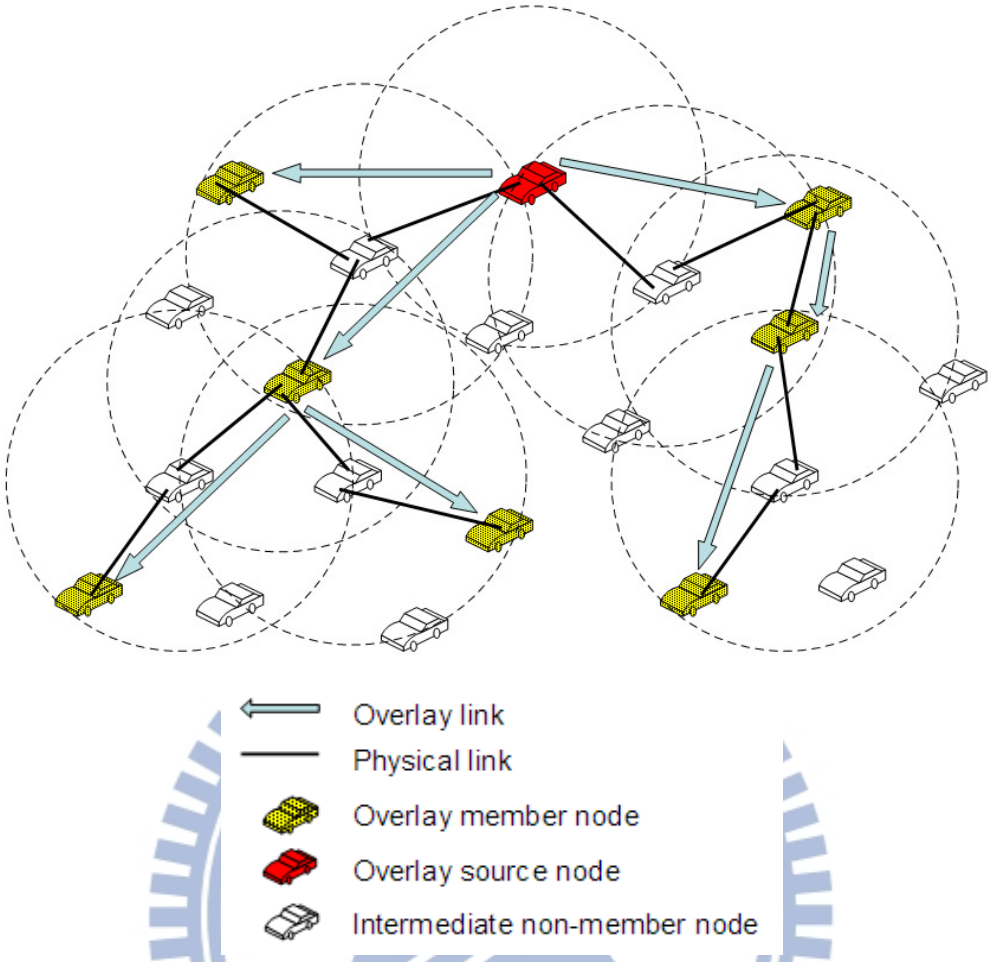


Figure 2. An example overlay multicast in a VANET.

1.1.2. QoS-satisfied dynamic overlay for urban VANETs

Using an overlay for live streaming in urban VANETs, due to that live streaming is delay-sensitive, the overlay should be able to deliver streams to each group node timely. In urban VANETs, facing many obstacles and road intersections, the connections between children nodes and their parents may suffer from frequent disconnections and result in high packet loss. Therefore, we propose two strategies to enhance the overlay stability: (1) QoS-satisfied dynamic overlay and (2) mesh-structure overlay. The QoS-satisfied strategy is proposed to dynamically adjust the overlay to meet its QoS. The mesh-structure strategy is to further enhance the overlay's connectivity.

As far as we know, only Qadri *et al*. studied an overlay for multimedia streaming in VANETs, but they uses a *static* overlay [1][11][24]. In MANETs, there existed literatures of dynamic overlays [16][17][18][19][25], but they were not designed for multimedia streaming. Facing high-mobility, obstacle-prone and broken-link-prone VANETs, they fail to meet the QoS requirements of multimedia streaming. Therefore, we propose a *dynamic* overlay for live multimedia streaming that focuses on QoS in terms of the stream's packet loss rate and end-to-end delay. It is a simple and effective QoS-satisfied overlay approach to cope with high mobility in urban VANETs.

The QoS-satisfied overlay strategy is for a group node to switch to a new parent which can offer better QoS. This strategy is receiver oriented. To achieve feasible streaming quality, a low packet loss rate and timely receiving of data packets is required. Therefore, we propose to use both the stream's packet loss rate and end-to-end delay to decide whether to switch to a better parent. Finally, to further ensure packet arrival's reliability, a two-parent mesh overlay is constructed. If the route to one parent temporarily fails, the child can still receive streaming data from the other parent.

In addition, in urban VANETs, there are full of obstacles [13][23][31]. Our simulation results show that the packet loss rate under large obstacles, such as buildings, can be very high compared to that under no large obstacles. Therefore, we also investigate a feasible stream rate and an overlay size for a city map of different road section sizes.

In summary, we propose a dynamic overlay multicast design suited for multimedia streaming in urban VANETs. To the best of our knowledge, dynamic overlay multicast for multimedia streaming has not been studied in VANETs literature.

1.2 Routing in urban VANETs

Routing in urban VANETs is quite challenging due to high mobility of vehicles (nodes) and dense large obstacles, such as buildings. Due to high mobility, communication links may be broken easily. And dense obstacles limit the radio coverage of a node within a road section, which degrades node connectivity. As a result, the routes among nodes in VANETs are quite instable. However, node movements are restricted by the road geometry so that node movements can be predicted. Besides, vehicle engines can provide unlimited energy for computation. Nodes may fully utilize the information they learn from each other to obtain optimal routing decisions. Therefore, routing protocols for VANETs can be well designed to adapt to the characteristics of VANETs, especially in urban areas.

In urban VANETs, traditional routing protocols are node-based [54][55], while most of recent routing protocols are road-based [45][46][47][48]. In node-based routing protocols, a path is composed of selected nodes, and packets are relayed along the determined sequence of the nodes. However, due to high mobility in VANETs, nodes move apart from each other easily. The link of two consecutive relay nodes may easily get broken so that route failures occur frequently in node-based routing protocols. Frequent route failures result in a high packet loss rate and extra control overhead for route re-discovery. Therefore, road-based routing protocols were proposed to improve route stability. In road-based routing protocols, a path consists of a succession of road sections (RSs) from source to destination, and data packets are transferred along the RSs. Within each RS, geographical forwarding is used. Geographical forwarding is that upon receiving a packet then a relay node decides its next hop depending on current neighbor nodes' positions. The neighbor node currently nearest to the next RS would be chosen as the next hop. In this way, the choice of a next hop is flexible. As long as nodes are dense enough in each RS of a path, a data packet can successfully be transferred hop by hop until reaching the destination. In urban VANETs, node density in an

RS does not vary too much, in a short period of time. Therefore, road-based routing is steadier than node-based routing.

To further enhance route stability, multipath routing provides alternative routes immediately once the current route fails [42][43][49], or provides concurrent transmission with multiple paths [44]. However, existing multipath routing approaches [42][43][44][49] are node-based. They still inherit the drawback of frequent routing failure in node-based routing protocols.

On the other hand, to find the routes with better quality (e.g., higher probability of connectivity, longer path lifetime, lower packet delay, etc.), some QoS routing approaches for urban VANETs were proposed [45][51][52][53]. However, most of current QoS routing protocols for VANETs are node-based, and they only consider scenarios of straight roads (e.g., highways) or city roads with limited scope of one or two intersections. For a general city geometry, road-based routing is more suitable. Recently, IGRP [45] was proposed to utilize node density and average speed in each RS so as to calculate the probability of connectivity and hop count, etc., from source to destination. However, it needs an additional traffic statistics service to obtain the required node density and average speed information.

In this dissertation, we propose a road-based QoS-aware multipath routing protocol for urban VANETs. Since road-based routing has been shown suitable in urban VANETs [45][46][47][48], we extend it to find multiple road-based paths. In addition, with multiple paths available, we propose a QoS path switching scheme to efficiently utilize a better path to achieve better QoS. Due to node mobility, for each RS in a path, the RS becomes connected or disconnected as time elapses. Therefore, we propose a space-time planar approach to predict each RS's future connectivity in a path, and then proceed to derive a path's future life periods. With each path's future life periods, the source node may dynamically switch to use the path which is currently connected and has shorter delay so as to enhance data transmission

QoS. To the best of our knowledge, there is no literature on road-based multipath routing protocol for VANETs.

Chapter 2

Related work

Existing multimedia streaming approaches and routing protocols for urban VANETs are reviewed in this chapter. We also provide their classifications and comparison tables to compare different approaches.

2.1 Multimedia streaming approaches for VANETs

Existing literatures on multimedia streaming approaches for VANETs are classified in Figure 3. We review these literatures in the following, and discuss their feasibility of multimedia streaming to a group of nodes in urban VANETs.

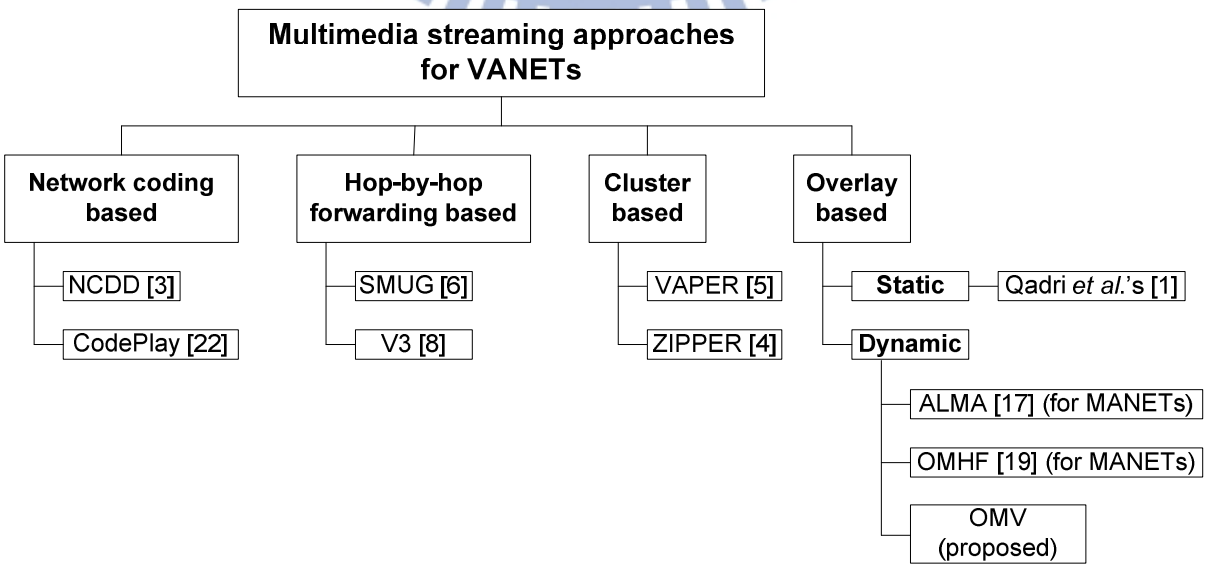


Figure 3. Classification of existing multimedia streaming approaches for VANETs.

2.1.1. Network coding based streaming

NCDD (Network Coding based Data Dissemination) [3] utilize random network coding techniques for data dissemination in VANETs. Each group node broadcasts its resource information to its 1-hop neighbors periodically. In addition, group nodes exchange coded pieces instead of original pieces. If a coded piece is linearly independent of the coded pieces in a node's local memory, then the node stores it. A node has to collect enough pieces then for decoding [2]. Note that network coding based approaches require group nodes periodically broadcast its collected pieces' information and retrieves uncollected pieces. Broadcast packets are not always received by neighbor nodes, and the concurrent transmitting nodes may suffer from severe collision [4]. Furthermore, network coding based approaches may consume a lot of time to collect enough pieces to decode if group nodes are not dense enough [4]. Especially in urban VANETs, due to obstacles, it is difficult for group nodes to hear the broadcasting of pieces information from one another. CodePlay [22] improves the collision problem of network coding based approaches (e.g., [2][3]) by adopting a local push scheme that only selected nodes are allowed to push data packets to other nodes in the same road segment. However, this technique still did not resolve the problem of group nodes not dense enough.

2.1.2. Hop-by-hop forwarding based streaming

In SMUG (Streaming Media Urban Grid) [6], a media stream is generated from a certain point (e.g. a roadside access point), and the stream is fed to SMUG-capable nodes and is distributed across a VANET [6]. Each node may dynamically be selected as a forwarder, and its transmissions are scheduled according to a TDMA scheme. Each forwarder is scheduled in a certain time slot to transmit, and neighboring forwarders would be assigned different time slots according to the proposed graph coloring technique so as to minimize the chance of collisions in adjacent areas [6]. However, if SMUG-capable nodes are not dense enough, it

would result in high packet loss, due to hard to meet any forwarder around. Besides, SMUG can only be applied in TDMA-based ad hoc networks, and it requires all nodes to follow its specific TDMA channel access scheme.

V3 [8] provides a scheme to retrieve the scene of a certain area to an interested vehicle. The application scenario of V3 is that for a certain region on the road, the scene can be captured by one or more video sources, such as pre-deployed stations or vehicles passing by. The interested vehicles, called receivers, continuously trigger the video sources to send the videos back. However, this scheme is not suitable for group communications, because each receiver establishes a path to a source, which is inefficient. Besides, the packet forwarding protocol in V3 only considers vehicles in a straight road, such as a highway. Therefore, V3 is not feasible to urban scenarios where a road map has many road intersections.

2.1.3. Cluster based streaming

Both VAPER (Vehicles Adaptive Peer-to-peer Relay Method) [5] and ZIPPER (Zero-Infrastructure P2P System) [4] form clusters among vehicles, and multimedia stream are relayed between clusters. Every vehicle periodically sends a beacon to neighbors to form clusters. There are a cluster head and also a cluster tail in a cluster. Each vehicle in the same cluster is one hop neighbor to each other. The main difference between VAPER and ZIPEER is that VAPER pushes a multimedia stream, while ZIPPER pulls a multimedia stream. In VAPER, the cluster head broadcasts a multimedia stream to its cluster members, and then the cluster tail relays the multimedia stream to the cluster head of the subsequent cluster. ZIPPER assumes a multimedia stream is composed of blocks, and a vehicle can retrieve blocks from other vehicles if available. If a required block is found, the block would be sent back. However, both clustering schemes require all vehicles to form clusters and maintain the clusters all the time, regardless of whether cluster members want to receive multimedia

stream. And both clustering schemes consider only straight roads, such as highways. They did not consider urban scenarios where the map has many road intersections. That is, the clustering schemes can not be directly applied to urban VANETs.

2.1.4. Overlay based streaming

In overlay based streaming, the source node multicasts a multimedia stream to group nodes in an overlay. Various kinds of overlay multicast approaches over MANETs have been proposed [16][17][18][19][25], while in VANETs, only Qadri *et al*. [1][11][24] discuss video streaming using a static overlay, as far as we know. Due to high-speed, obstacle-prone and broken-link-prone characteristics, urban VANETs is very different from MANETs. For dynamic overlay approaches proposed for MANETs, most of them [16][18][25] aim to maintain low cost topology in terms of number of overlay links or physical links, but they cannot adjust the overlay in time for urban VANETs with high mobility. That is, they may cope with MANETs with low mobility; however, they are not feasible to urban VANETs with high mobility. Instead of maintaining low cost topology, OMHF [19] and ALMA [17] adjust an overlay quickly based on current overlay links' quality. OMHF [19] uses the number of a node's link failures to indicate its quality. If a parent's link quality is lower than its child's, their parent-child roles are exchanged. However, OMHF is not suitable for obstacle-prone urban VANETs. In obstacle-prone VANETs, the new parent may not be able to connect to the ancestor, and some children may not be able to connect to the new parent. ALMA [17] uses the overlay link delay as the estimated quality of an overlay link. If the link delay of a child to its parent exceeds a threshold, the child switches to another parent with shorter link delay. However, if applying ALMA to urban VANETs, due to high packet loss and frequent disconnections, a child may switch to a parent with higher packet loss, although the link delay is low. Therefore, the existing approaches proposed for MANETs are not feasible to

multimedia streaming in urban VANETs. Feasible multimedia streaming in VANETs needs to consider the connectivity of children and parents as well as the stream's packet loss rate and the end-to-end delay.

As to the related work in VANET, only Qadri *et al*. [1][11][24] discussed video streaming using an overlay, as far as we know. They evaluated the feasibility of video streaming using a *static* overlay in VANETs, and showed the improvement of applying video coding and error resilience. However, the overlay structure was static, even though nodes are mobile, which may suffer from inefficient overlay structures and frequent disconnections. In our work, we focus on dynamic overlay adaptation. In Table 1 the aforementioned approaches are compared, considering the feasibility of multimedia streaming for a group of nodes in urban VANETs.

Table 1. Comparison of existing multimedia streaming approaches for VANETs.

Approach	Hop-by-hop forwarding based		Network coding based [2][3][22]	Cluster based [4][5]	Overlay based		
	V3 [8]	SMUG [6]			Qadri *et al.*'s [1]	ALMA [17]	OMV (proposed)
Basic idea	Continuously trigger and reply between a receiver and the source	Scheduled multicast	Network coding coded packets are exchanged among members	Nodes form clusters, and packets are relayed between clusters	Static mesh overlay (only 7 nodes)	1. Dynamic tree overlay 2. Overlay is adjusted according to overlay link delay (designed for MANETs)	1. Dynamic tree / mesh overlay 2. Overlay is adjusted according to packet loss rate and end-to-end delay
Suitable for obstacle-prone environments (eg. urban)	No	Yes	No	No	Yes	Yes	Yes
Required density of group nodes	High	Medium	Medium	High	Low	Low	Low
QoS supported	No	No	No	No	No	No	Yes
Obstacles evaluated	No	No	No	No	Yes	No	Yes

2.2 Routing protocols in urban VANETs

Existing routing protocols for urban VANETs are classified as Figure 4. The routing approaches of each category are reviewed in the following subsections.

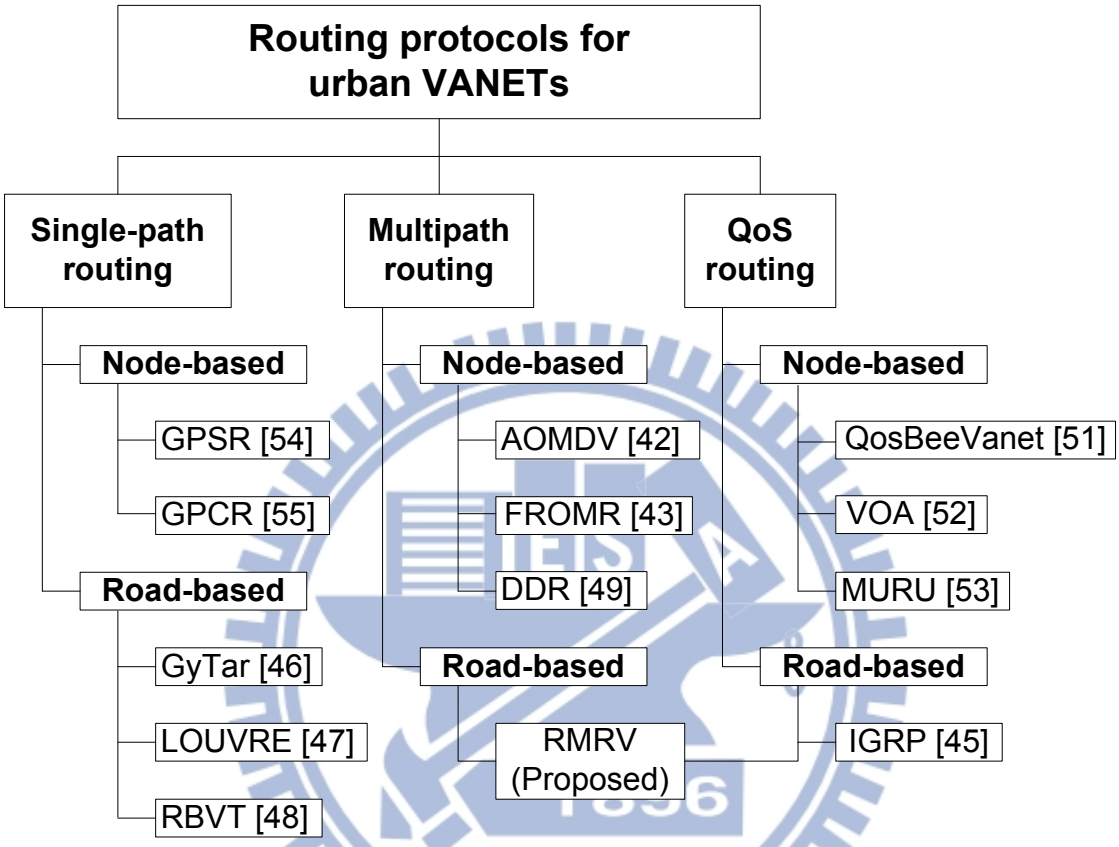


Figure 4. Classification of routing protocols for urban VANETs.

2.2.1. Road-based vs. node-based routing

In literatures of routing protocols for urban VANETs, vehicles are assumed to be equipped with GPS (global positioning systems) and street-level city maps [45][46][47][48]. With GPS, each node knows its own and its neighbors' positions, and may utilize the position information to forward data packets toward destination [54][55]. These kind of routing protocols are node based. However, due to dense obstacles in city maps, sometimes a routing path may fail to be found. Therefore, road-based routing protocols are proposed. GyTar (improved Greedy Traffic Aware Routing protocol) [46] is an intersection-based geographical

road-based routing protocol. A relay node computes the score of each neighboring road intersection, and decides the next road intersection to forward packet to. The score is decided by the node density and road distance to the destination. However, GyTar is a greedy algorithm so that the current chosen road intersection may later lead to road sections with poor node density and connectivity.

LOUVRE (Landmark Overlays for Urban Vehicular Routing Environments) [47] builds an overlay on top of an urban topology [47]. The overlay consists of connected roads in the topology. To form an overlay, LOUVRE uses a P2P protocol for discovering and distributing traffic density information of each road among all nodes in a network environment. Each node exchanges traffic density information with its neighbors via broadcasting. Then, a path between a source and a destination on the overlay can be found by using Dijkstra's algorithm. However, the scalability of the overlay is limited due to P2P exchange delays in the network environment.

The RBVT (Road-Based using Vehicular Traffic) routing protocol [48] creates a road-based path consisting of a succession of road intersections [48]. The authors proposed a reactive protocol, RBVT-R and a proactive protocol, RBVT-P. The route discovery scheme in RBVT-R is similar to the traditional source routing protocol, but the routing information piggybacked in a packet is a succession of road intersections instead of nodes. RBVT-P is similar to LOUVRE. Nodes periodically discover and disseminate the road-based network topology to maintain a global view of the network connectivity. The performance of RBVT-R depends on the stability of every RS's connectivity, and the performance of RBVT-P depends on the delay of network connectivity information exchange.

2.2.2. Multipath routing

There is no existing road-based multipath routing protocol for VANETs, as far as we know. In the following, we review existing node-based multipath routing protocols for urban VANETs. AOMDV [42], extended from AODV [59], is a representative multipath routing protocol for MANETs. If it is applied to VANETs, due to high mobility, link lifetimes to next hops expire soon, thus an alternative route may still fail at a certain relay node. FROMR (Fast Restoration On-demand Multipath Routing) [43], which is a node-based routing protocol, extends AODV to discover multiple paths so as to rapidly build an alternate path if the current route fails. Unlike AODV where duplicate copies of RREQ messages are simply discarded, FROMR uses some of these copies to form multiple reverse paths. In addition, to reduce control overhead, FROMR partitions a geographic region into grids and assigns each grid a grid leader. Only leaders are responsible for control messages broadcasting and data forwarding. Although FROMR provides more choices of next hops, in essence the link lifetimes of these next hops still expire soon, due to high mobility in VANETs. DDR (Differentiated Reliable Routing) [49] is also a multipath routing protocol in hybrid VANETs, which have road side units (RSUs). In hybrid VANETs, a multi-hop path may include RSUs as intermediate nodes for packet relaying.

2.2.3. QoS routing

Most of existing QoS routing protocols in VANETs are node-based. QoSBeeVanet [51] imitates honey bees' behavior to find a route with better QoS in VANETs. VOA [52] assumes a highway scenario. It selects an optimal routing path and backup routing paths in order to gracefully switch to a backup routing path before the current routing path is broken. MURU [53] uses an expected disconnection degree (EDD) to estimate a probability that a route will fail during a given time period. MURU utilizes EDD to obtain an optimal path. But MURU

only utilizes local map information and it may lead to the local optimum problem.

IGRP (Intersection-based Geographical Routing Protocol) [45] is a road-based single path QoS routing protocol. In IGRP, a path is composed of a succession of road sections (RSs). Therefore, a path's QoS may be estimated with the QoS of RSs. IGRP first retrieves the node density and average node speed of each RS using an existing location service. Then, these data are utilized to probabilistically estimate QoS related metrics, such as probability of connectivity, end-to-end delay, hop count, and bit error rate (BER). Therefore, for paths between source and destination, only the path that satisfies the constraints of these metrics is selected.

Finally, in Table 2, we summarize the aforementioned road-based routing protocols, including their basic classifications, main idea and limitations.

Table 2. Comparison of representative road-based routing protocols for urban VANETs.

Approach	Single path or multipath	QoS-aware	Main idea	Limitations
GyTar [46]	Single path	No	Compute the score of each neighboring road intersection with node density and distance to destination	As a greedy algorithm, the current chosen road intersection may later lead to RSs with poor node density and connectivity
LOUVRE [47]	Single path	No	Routing path is decided according to the overlay which is built on top of urban topology, consisting of connected roads in the network	Scalability of the overlay is limited due to the road density information exchange delay in the whole network
RBVT [48]	Single path	No	RBVT can find a source-destination path consisting of RSs. Then every packet is relayed along the routing path, according to the RS list piggybacked in the packet's header	RBVT-R's performance depends on the stability of each road section's intra-connectivity; RBVT-P's performance depends on the delay of network connectivity information exchange
IGRP [45]	Single path	Yes	Probabilistically formulate connectivity, delay and bit error rate to derive a path's QoS	An additional traffic statistics service is required for the probability calculations of road connectivity, BER, etc.
RMRV (proposed)	Multipath	Yes	Find multiple paths and predict their future life periods for QoS path switching, with very little overhead	If nodes are dense, the advantage of providing multiple paths is limited

Chapter 3

Proposed dynamic overlay multicast for live multimedia streaming in urban VANETs

3.1 Overview of the proposed overlay multicast in VANETs (OMV)

This chapter considers the scenario of live multimedia streaming multicast to vehicles of the same group, and these group-member vehicles are organized as an overlay tree, as shown in Figure 5(a) and Figure 5(b). Group-member vehicles (called group nodes or overlay nodes, interchangeably) join the multicast tree initiated by a vehicle which is a streaming provider; this vehicle acts as the multicast source node. Multimedia packets generated by the multicast source node are delivered along the overlay tree. The link between a parent and its child may be actually a multiple-hop path through intermediate non-member nodes, which is referred as an overlay link. In an overlay link, packets are passed using the underlying network-layer routing protocol. The source node also plays the role of the bootstrap node (BSN) of the multicast tree. When a node wants to join the multicast tree, it firstly asks the BSN for which node to be its parent, so as to become a member of the tree.

As vehicles move rapidly, an overlay tree need to change its structure to maintain QoS. For example, as shown in Figure 5(a), node 4 moves fast toward the right hand side, and it is

going far way from its parent node 3 and child node 5. Such a situation will result in high packet loss and long packet delay in the overlay. Therefore, we propose to dynamically adjust the overlay to maintain its QoS. In addition, to further enhance the robustness of an overlay, we build a two-parent overlay, as shown in Figure 5(c). We propose this dynamic overlay (i.e. QoS-satisfied dynamic overlay and two-parent overlay) to suit for live multicast streaming scenarios, so that passengers in group-member vehicles can enjoy live reliable video streaming from the source vehicle.

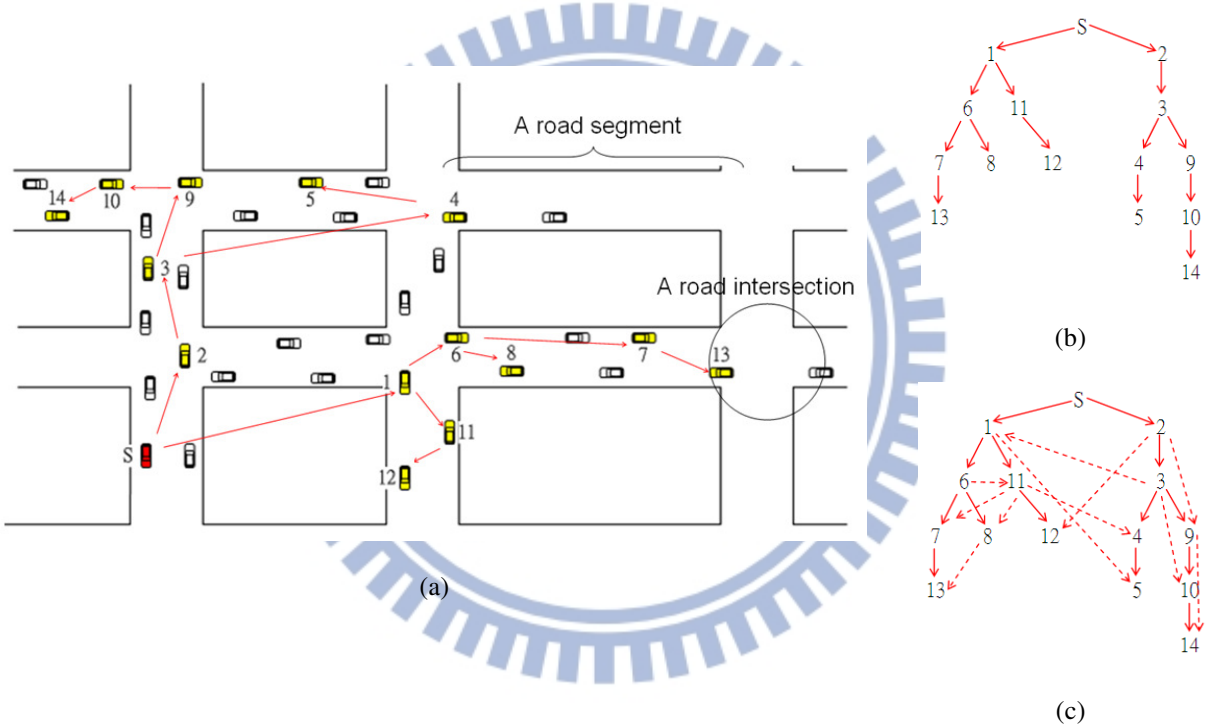


Figure 5. An illustration of a multicast overlay in urban VANETs: (a) an overlay formed by interested vehicles, in urban roads; (b) the corresponding overlay tree; (c) expanding the tree to a mesh, so that nodes are allowed to have two parents.

3.2 Proposed overlay construction and QoS-satisfied overlay adaptation

In the following, we introduce the proposed overlay construction and QoS-satisfied overlay adaptation. Firstly, we present the life cycle of a group node in a multicast overlay in Figure 6. The life cycle of a group node includes joining the multicast overlay, periodically exchanging its movement state table with other group nodes, dynamically adjusting its parent, and finally choosing to leave the multicast overlay.

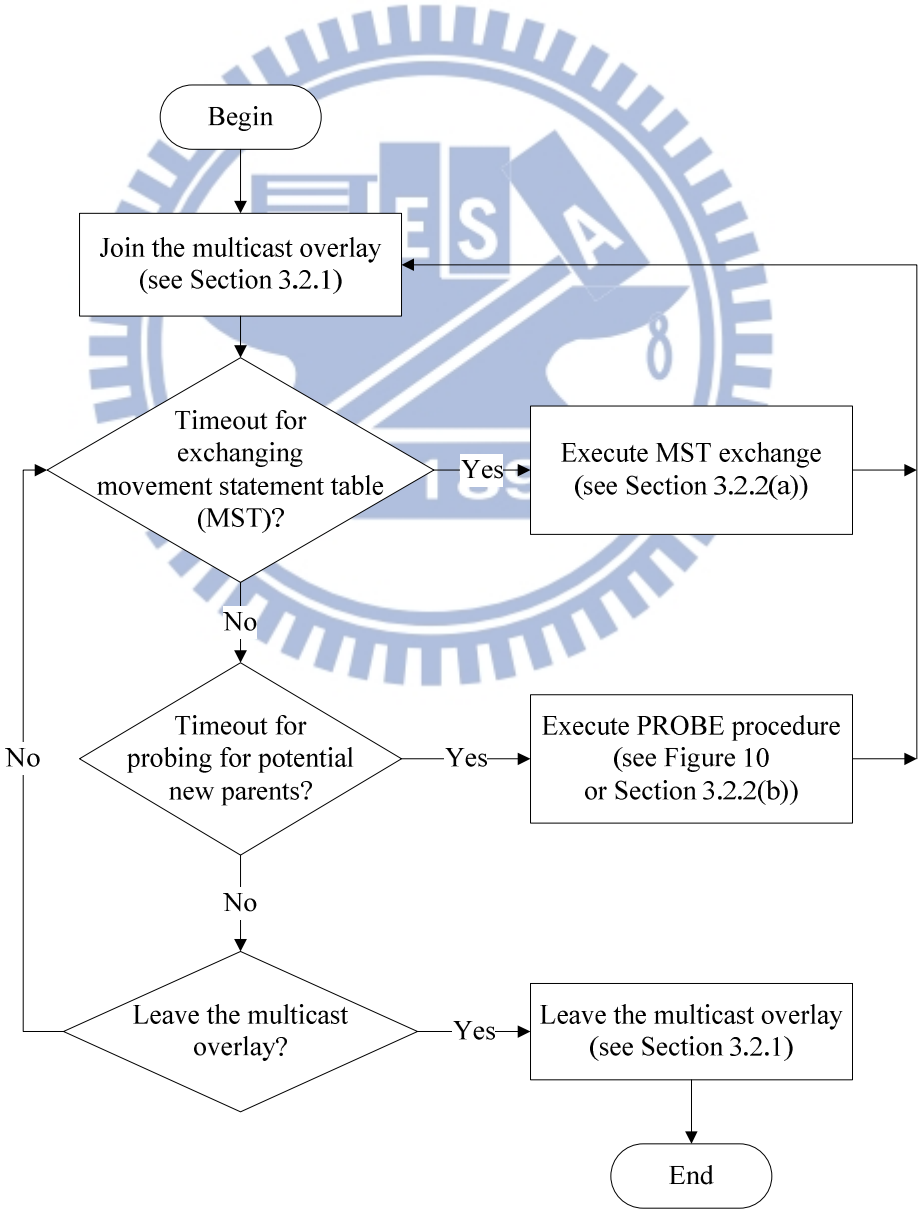


Figure 6. Flowchart of the life cycle of a group node in the proposed multicast overlay.

3.2.1. Join and leave

As mentioned previously, the streaming source node also acts as the BSN of the multicast tree. The BSN has an *overlay structure table* which records the current successfully joined nodes, their coordinates, and their children. The purpose of maintaining the overlay structure table in the BSN is to select a candidate parent for a new node requesting to join the multicast overlay. Besides, the BSN periodically broadcast its node ID to other nodes interested in joining the group so that the node ID of the BSN is available to all interested nodes. When a node requests to join an overlay, the join procedure is executed, which includes five stages: (1) a requesting node sends a JOIN request to the BSN to ask which node to be its parent. (2) Then, from the BSN's overlay structure table, the BSN chooses a node as the candidate parent and replies to the requesting node. A node in the same road section as the requesting node is the best choice as its parent; otherwise, the node nearest to the requesting node would be chosen. (3) When the reply from the BSN is received, the requesting node sends a JOIN_PARENT request to the candidate parent. The candidate parent will accept it if its number of children does not exceed a given maximum number of children (MAX_CHILDREN). (4) If the JOIN_PARENT request is rejected, the requesting node asks the BSN again and executes the join procedure (2) ~ (4) until it successfully finds a parent. (5) When the requesting node successfully joins the parent, it then informs the BSN to update the overlay structure table. Note that in step (4), the requesting node also informs the BSN of the candidate parent that it failed to join. Then in step (2), the BSN would avoid choosing this candidate parent.

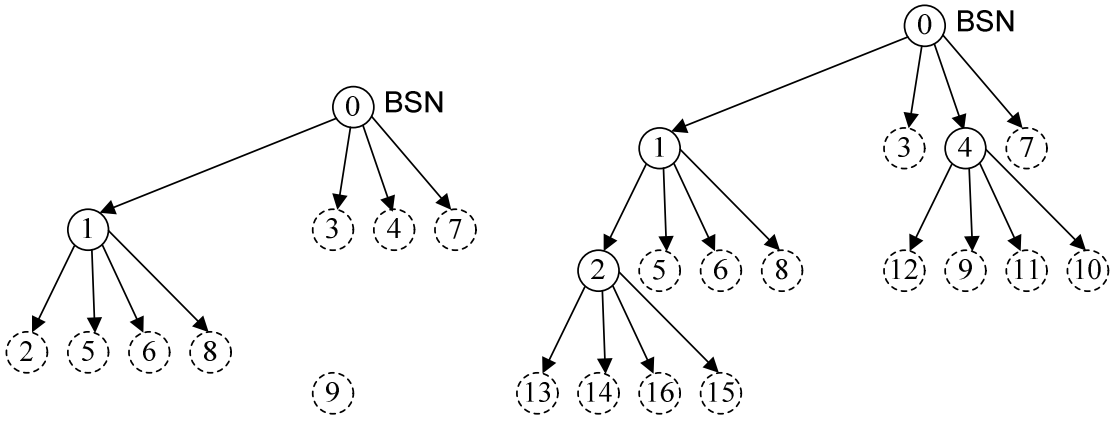
Note that there is a situation that rarely happens in MANETs, but we found it common in urban VANETs. The earlier successfully joined nodes may be forced to accept too many children nodes. Due to route discovery being more difficult in urban VANETs, it often takes 5 to 10 seconds or more for a node to connect to the BSN successfully. And so does to join the

candidate parent. Much time is wasted in request retries. This situation not only relates to end user satisfaction, but it may cause imbalance load. That is, when an interested node requests to join the multicast overlay, it usually takes long time to complete the join procedure in urban VANETs. As a result, the earlier successfully joined nodes may be forced to accept too many children nodes. Figure 7 shows an example of an imbalanced tree. Assume node 0 is the BSN. The node number is assigned according to the order of the BSN receiving the corresponding node's JOIN request. For example, node 3 is the third node that the BSN receives a JOIN request. A dotted node means that it has not finished the join procedure. Initially, node 1 successfully joined node 0. At this time, only nodes 0 and 1 can be candidate parents. Then, in a short duration, the BSN receives nodes 2 ~ 9's JOIN requests. And the BSN replies each requesting node a candidate parent of node 0 or node 1, except for node 9 because the children number of either nodes 0 and 1 has reached MAX_CHILDREN (e.g., 4). Node 9 has to wait for a new candidate parent. As the BSN is handling the JOIN requests, nodes 2 and 4 finish the join procedure, as shown in Figure 7(b). At the same time, the BSN has received nodes 10 ~ 15's JOIN requests. So the BSN replies nodes 9 ~ 15 that their candidate parent are nodes 2 or 4. Note that node 0 and node 1 cannot be candidate parents because they have reached MAX_CHILDREN. Therefore, as we can see, the earlier successfully joined nodes may be forced to accept too many children nodes. This is due to that it takes long time for a node to finish the join procedure, and by then there may have several arrived nodes waiting for joining the overlay.

A solution is as follow. Regardless of requesting nodes successfully joining the given candidate parents or not, all requesting nodes can be used by the BSN as other nodes' candidate parents so as to increase the number of available candidate parents, as shown in Figure 8(a). However, in urban VANETs, route failures are common. For example, node 5 may fail to join node 3. Until node 5 finds a parent, the whole subtree (nodes 4 ~ 9) of nodes 5

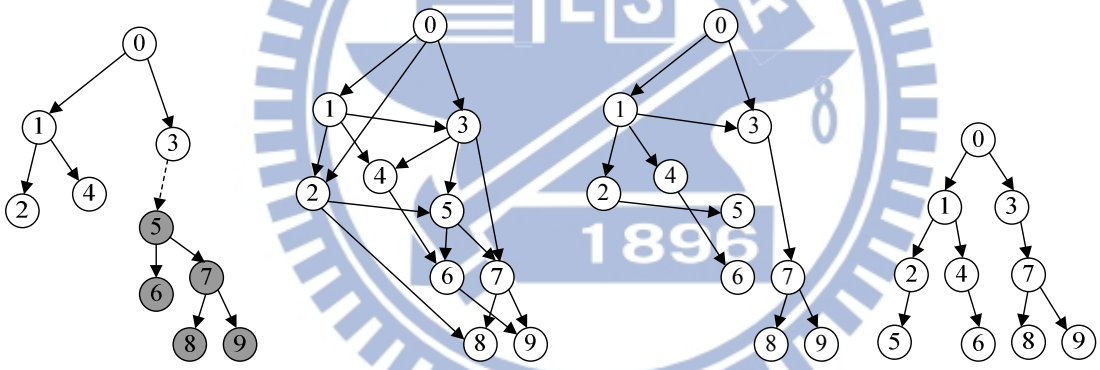
cannot start the multimedia streaming service. To conquer this, we propose a simple and effective solution. We allow the BSN to provide two candidate parents to a child at once in stage (2) of the JOIN procedure. The resulting overlay structure is shown in Figure 8(b), which is a two-parent mesh overlay. In this way, a requesting node has higher possibility to join a parent successfully. For a tree overlay, when a requesting node successfully joins one of the candidate parents, the JOIN request to the other candidate parent will be canceled. Figure 8(c) shows the final tree, and is redrawn as shown in Figure 6(d). Thus, the imbalanced tree problem is resolved, and the join delay is shortened as well.

Finally, when a node wants to leave the multicast tree, it sends a LEAVE message to its neighbors so as to handover its children to its parent. The LEAVE message lists the leaving node's children's IDs and its parent's ID. The format of the LEAVE message is <*node ID*, *parent's ID*, *number of children*, *child 1's ID*, *child 2's ID*, …>, so that each child knows its new parent, and the parent knows its new children. Note that to avoid a leaving node's parent from becoming overloaded, the leaving node's parent will only accept new children up to its maximum number of children (MAX_CHILDREN). The leaving node's parent accepts new children that are close to it based on its movement state table. For those children not accepted by the leaving node's parent, they will find their new parents via a parent selection procedure, called PROBE. The detail of the PROBE procedure is described in the following Section 3.2.2(b).



(a) (b)

Figure 7. Imbalanced tree problem due to JOIN requests bottleneck in the BSN. (a) The BSN receives nodes 2 ~ 9's JOIN requests, but currently only nodes 0 and 1 can be candidate parents. (b) Nodes 2 and 4 finish the join procedure. Then the BSN replies nodes 9 ~ 15 of their candidate parents as nodes 2 or 4.



(a) (b) (c) (d)

Figure 8. Solution to the imbalanced tree problem, as shown in Figure 7(a). (a) All requesting nodes are allowed to be assigned by the BSN to be candidate parents. Note that join failure of a whole subtree may occur, such as node 5 and its descendants. (b) In addition to (a), let the BSN provide two candidate parents to a child at once. (c) The candidate parent which is slower to finish the join procedure will be released. (d) We have a more balanced tree.

3.2.2. QoS-satisfied overlay adaptation

To dynamically adjust an overlay structure so as to maintain a low packet loss rate, we propose a QoS-satisfied overlay adaptation mechanism by choosing a better parent dynamically. An example is shown in Figure 9. Assume node A moves far away from its parent, resulting in frequent packet loss. At this time, node B is close to node A. Node B may be more appropriate to be node A's new parent. And so does node C. Node A has to decide whether to switch to a new parent that may provide a better QoS. Here we propose a PROBE procedure for this purpose. However, some knowledge about nodes' positions (coordinates) is required in the PROBE procedure. So we first describe how overlay nodes exchange their positions in Section 2(a), and then introduce the PROBE procedure in Section 3.2.2(b).

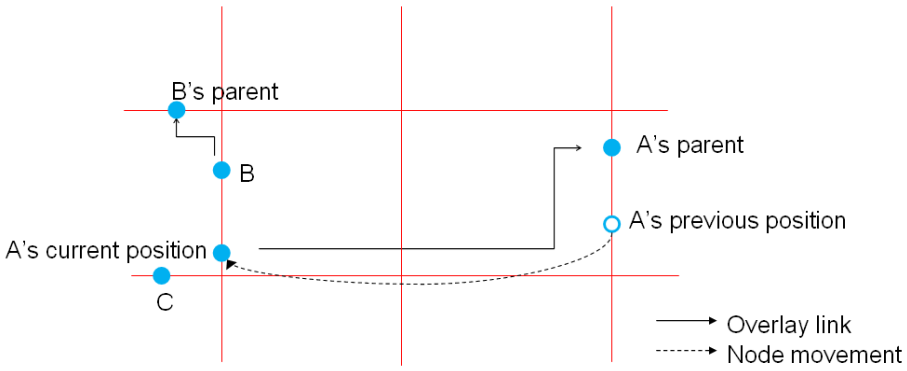


Figure 9. Choosing a better parent.

(a) Movement state table exchange

In the proposed OMV, each overlay node maintains a movement state table (MST) that records the movement state of each overlay node. A movement state contains <*nodeID*, *x*, *y*, *parent's ID*, *timestamp*>, where *x* and *y* are coordinates of node *nodeID* obtained by GPS. Each overlay node periodically exchanges its MST with its overlay neighbors. Note that each overlay node only exchange MSTs with its overlay neighbors so as not to incur too much traffic load. Before a node sends its MST to its neighbors, it refreshes its own movement state in the MST. When an overlay node receives an MST from a neighbor, it updates its own MST

with the received MST.

(b) Probing for a new parent

As shown in Figure 9, node A may have multiple potential new parents. A new parent with low packet loss and low end-to-end delay is most preferred. In the following, we propose a PROBE procedure to decide whether to switch to a new parent for better QoS. Note that to reflect the correct packet loss rate of a node, the calculation of the packet loss rate only bases on the duration from the time the node switching to its current parent to the current time. In addition, for live streaming, each node has a playout buffer. Thus, we define the *packet loss rate* of node$_i$ as:

Let t = the time when node$_i$ switched to its current parent.

$Dr_i(t)$= non-duplicate data bytes node$_i$ received, from time t to the current time t_c. Packets not arriving within playout buffer time are discarded.

$Ds(t)$ = data bytes of the stream, from time t to the current time t_c. Thus $Ds(t)$ = source stream rate $\times$ $(t_c - t)$.

Therefore, the *packet loss rate* r_i of node$_i$ is 1 - Dr_i/Ds.

For example, supposing the source stream rate is 128 *kbps*. If node$_i$ switched to its current parent at time 100 *s*, and node$_i$ receives a PROBE request at time 120 *s*, then Ds = 128 *kbps* $\times$ (120 *s* – 100 *s*) = 320,000 *bytes*. Supposing during this period of 20 *s*, node$_i$ has received 240,000 *bytes*, then node$_i$'s packet loss rate = 1 - 240,000/320,000 = 0.25.

We also calculate the *end-to-end delay* for each overlay node. The *end-to-end delay* is the average time taken for a data packet to be transmitted from the source node to an overlay node. Note that the calculation of end-to-end delay only bases on the duration from the time a node switching to its current parent to the current time. In addition, there is no retransmission for packet loss, since we focus on live streaming. Duplicated data packets and data packets not arrived within playout buffer time are excluded in the calculation of end-to-end delay. The

end-to-end delay is used to examine that if an overlay node switches to a new parent, whether the associated end-to-end delay, from the source to the new parent to the overlay node can meet the playout buffer time requirement. In this way, a node will not select a new parent such that the node's end-to-end delay exceeds the playout buffer time.

Proposed PROBE procedure

As shown in Figure 10, assume that node A is to execute the PROBE procedure. From the MST, node A knows the other nodes' positions and which nodes are its descendants. From node A's MST, node A chooses the nodes which are within the distance threshold PROBE_RANGE (e.g. 500 meters) from node A. These nodes form a set P, such that size(P) $\leq$ MAX_PROBE. Note that the nodes near to node A would be chosen first. In addition, to avoid forming loops, the nodes which are node A's descendants would be removed from P. Then, node A sends a PROBE request to each node of P. Afterward, node A sets a timeout of T_W and wait for nodes in P to reply. For each node i that has received a PROBE request, it replies node A with its *packet loss rate* r_i, and its *end-to-end delay* d_i. Note that by GPS the clocks of all nodes can be synchronized. Therefore, the overlay link delay t_i, which is the delay from node i to node A, can be calculated using the timestamp in a PROBE reply packet. During T_W, node A may receive multiple replies from nodes of P. These nodes form a set R. We pick up the nodes from R such that the resulting end-to-end delay ($d_i + t_i$) can meet playout buffer time T_B for node A. These nodes form a set R'. That is, $R' = \{x_i \mid d_i + t_i < T_B, \forall x_i \in R\}$. Finally, we select the node z with the lowest packet loss rate from R' to be the candidate parent of node A. Then, node A sends a JOIN_PARENT to node z. If node z accepts the offer, node A sends a RESIGN message to node A's current parent. If node z rejects the offer, node A removes node z from R' and chooses another node with the next lowest packet loss rate from R' to be its candidate parent. If node z is the current parent of node A, node A

will not take any action.

Figure 11 uses an example to illustrate the PROBE procedure. Node A looks up its MST and found 5 nodes (nodes 1 ~ 5) are within PROBE_RANGE to itself. Assume MAX_PROBE is 3; thus, only three nodes would be chosen for probing. Therefore, nodes 2, 3, 4 will be chosen because they are closer to node A than nodes 1 and 5. Node A sends a PROBE request to each of these nodes. Then, each of them replies its packet loss rate r_i and end-to-end delay d_i. The overlay link delay t_i is also known using the timestamp in the PROBE reply packet. Assume the playout buffer time T_B is 3 *s*. For nodes 2, 3 and 4, node 2 cannot meet playout buffer time constraint, due to that $d_2 + t_2 > T_B$. Therefore, node A discards node 2. As to nodes 3 and 4, node 4 has a lower packet loss rate than that of node 3. Therefore, node A changes it parent to node 4 by sending a JOIN_PARENT to node 4.

Notice that a node's packet loss rate results from its parent's packet loss rate as well as the parent-child link loss rate. The parent's packet loss rate is addressed explicitly in the proposed PROBE procedure. In addition, the parent-child link loss rate is addressed implicitly in the PROBE procedure. If a child can successfully receive a PROBE reply from a probed parent before PROBE timeout, it implies that the associated parent-child link is reliable and has a low loss rate.

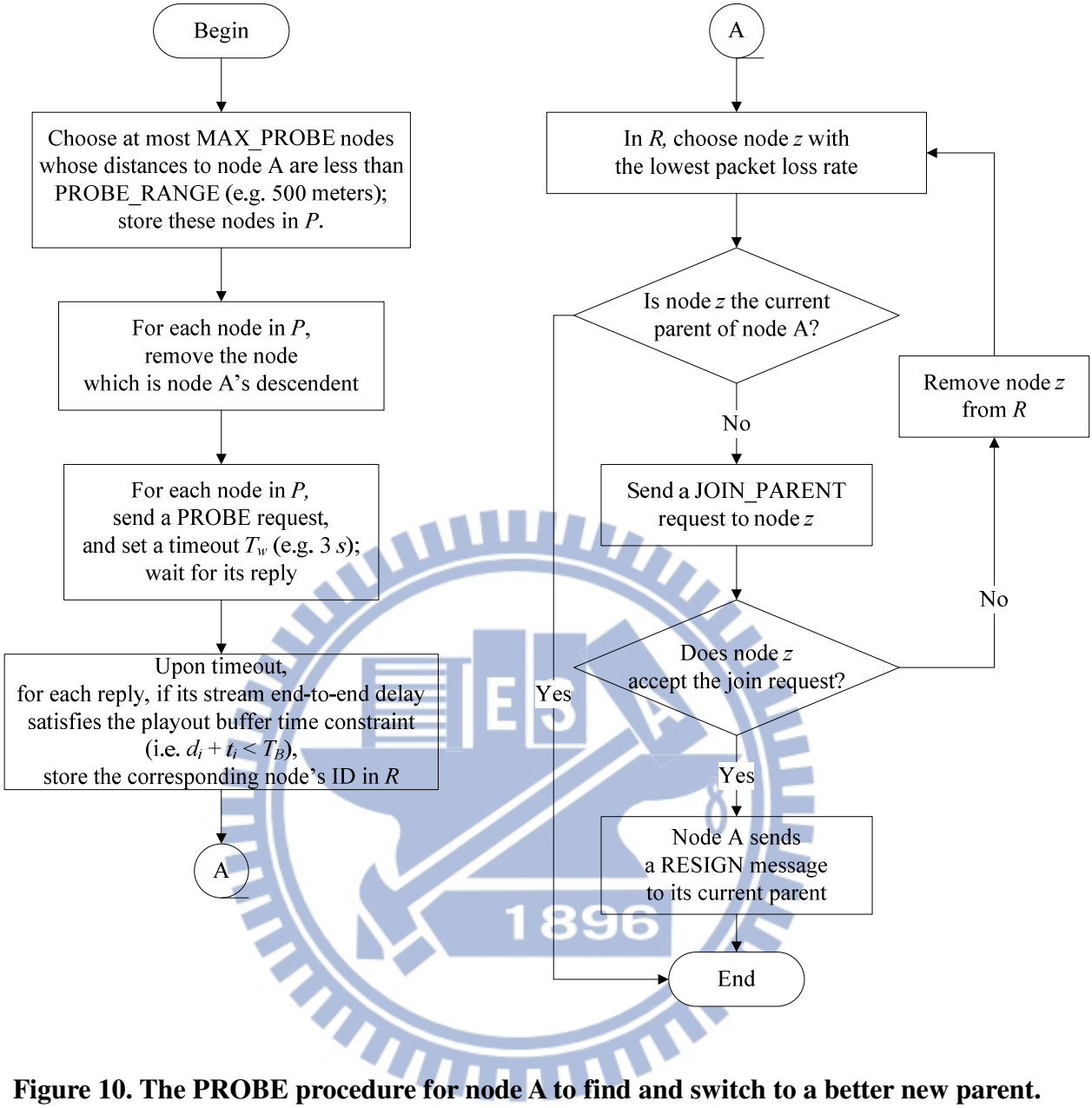


Figure 10. The PROBE procedure for node A to find and switch to a better new parent.

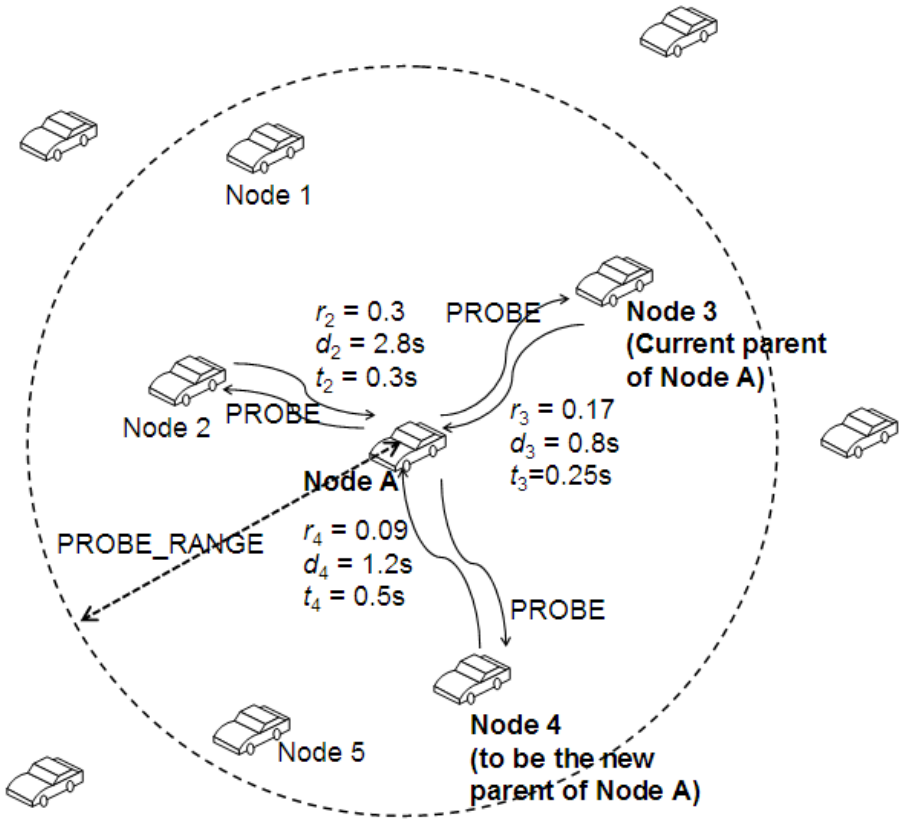


Figure 11. An example to illustrate the PROBE procedure.

Avoidance of hotspot nodes

Note that in the PROBE procedure, nodes with low packet loss rates may be more probable being selected as parents and may become hotspot nodes. Therefore, we set an upper bound (MAX_CHILDREN) of the number of children for a node. In addition, as a node accepting more children, its packet loss rate will increase due to more traffic load. As a result, the occurrence of hotspot nodes can be minimized.

Loop-free tree overlay

The loop problem may occur in an overlay due to two conditions: (1) a node chooses its descendant as its parent, or (2) two nodes simultaneously choose each other or each other's descendant as their parents [17]. For a tree overlay, these two conditions also result in network partitions. That is, a subtree is disconnected from the overlay. In fact, these conditions are

avoided in the proposed PROBE procedure. This is because a node always tries to switch to a new parent with low packet loss rate, which makes the overlay formed a *minimum heap*. That is, the packet loss rate of a node is lower than that of its descendants, in a minimum heap. Thus, for condition (1), a node would not choose any of its descendants as its parent, since the packet loss rates of the descendants are higher than that of the node. In addition, as mentioned previously in the PROBE procedure, a node will not choose its descendants and send them probe requests. This is to avoid loops and also not to waste time to probe for descendants. As to condition (2), in a minimum heap, for any node x, assuming node x chooses a new parent node y, this implies that the packet loss rates of node y and the ancestors of node y are lower than those of node x and those of node x's descendants. Consequently, neither node y nor the ancestors of node y would choose node x or node x's descendants as parents. That is, condition (2) would not happen in the PROBE procedure. In summary, the proposed tree overlay is a loop-free overlay, and it will not suffer from network partitions.

3.2.3. Two-parent overlay

The disconnection of any overlay node from its parent in a tree overlay leads to its entire offspring disconnected, which would result in severe performance degradation, especially in urban VANETs. In this dissertation, we use a mesh overlay to enhance the tree overlay. We adopt a two-parent overlay. Each node in an overlay may have two parents. If the route to one parent fails, the child can still receive streaming data from the other parent. For the mesh overlay, the procedures of MST exchange and PROBE are basically the same as those of the tree overlay. However, one thing is different from the tree overlay. In the PROBE procedure for the mesh overlay, if a node is going to replace its current parent with a better parent, the node does not resign the current parent immediately. The current parent is set as a "supporting parent," and the new parent is set as a "main parent." Only the main parent needs to do the

MST exchange and run the PROBE procedure. The purpose of the supporting parent is that if some stream is not received by the main parent, the child can still receive lost packets from the supporting parent. Duplicate packets will be discarded. If later the PROBE procedure finds a better parent, the supporting parent will be discarded, and the main parent will become a supporting parent. Finally, for the case of an overlay allowing more than two parents, our scheme is still workable: one main parent and the others are supporting parents. However, an overlay with too many parents is inefficient, due to too many duplicate data. In VANETs, we adopt two parents for a child in an overlay for a better packet loss and duplication data. Since main parents execute the proposed PROBE procedure, the tree formed by main parents is loop-free, as explained previously. Thus, the proposed mesh overlay would not suffer from network partitions.

3.2.4. Detection of a node unexpected leaving or a node disconnected from its parent

If a node unexpectedly leaves the overlay, or a node is disconnected from its parent, the overlay can be repaired by the PROBE procedure, as described in Figure 10. The PROBE procedure is periodically invoked. Therefore, for nodes disconnected from their parents, or for the children that their parent unexpectedly left, they will find new parents in the PROBE procedure.

3.2.5. Summary of our OMV features for urban VANETs

To adapt to urban VANETs, the proposed OMV scheme has the following features:

(1) Avoiding imbalanced load in the overlay structure: The earlier successfully joined nodes are often forced to accept more children nodes due to that route discovery has high

delay in urban VANETs. The BSN may become a bottleneck, and the tree overlay becomes imbalanced. So we avoid this problem in the proposed join procedure.

(2) QoS-satisfied dynamic overlay adaptation: Routing failures occur frequently in urban VANETs. To dynamically adjust the overlay structure so as to maintain the packet loss as low as possible, we propose a QoS-satisfied overlay adaptation mechanism that is incorporated in the PROBE procedure. A node periodically probe for a better parent with lower packet loss rate under the constraint of end-to-end delay.

(3) Multi-parent overlay: Facing high packet loss in urban VANETs, we allow each node in an overlay to have two parents, the main parent and supporting parent. If the route to one parent fails, the child can still receive streaming data from the other parent. In addition, the QoS-satisfied overlay adaptation is still executed by the main parent to dynamically adjust the overlay to meet its QoS.

3.3 Evaluation and discussion

To simulate real urban VANET environments, we adopted IEEE 802.11p transmission settings [21], an intelligent driving vehicle mobility trace generator [10] and city maps with large obstacles [13][23], in the simulations. We compare the proposed OMV with the best methods available, ALMA [17], which is a dynamic tree overlay multicast approach in MANETs and Qadri *et al.*'s work [1], which is a static mesh overlay in VANETs. Remind that to the best of our knowledge, there is no existing dynamic overlay multicast approach for VANETs in literature. Qadri *et al.*'s mesh overlay [1] is static and includes only 7 nodes. Here we extend it to include arbitrary overlay nodes in order to compare it with the proposed OMV for mesh overlays. We choose ALMA for comparison because it is a pure application-layer overlay multicast approach, just like the proposed OMV. Other existing dynamic overlay multicast approaches utilize cross-layer information, such as hop count, for overlay adaptation.

Since ALMA was designed for tree overlays, for fair comparison, we compare it with the proposed OMV for tree overlays. The performance improvements of the proposed OMV as well as its feasibility for multimedia streaming under different group sizes and road section sizes will be investigated.

We used QualNet 5.0 [20] to simulate the proposed OMV. Note that QualNet can fully simulate the communication behaviors in lower layers including PHY and MAC layers and queuing behaviors. The derived end-to-end delay includes propagation delay as well as transmission delay. According to the IEEE 802.11p [21], the transmission is in 5.9 GHz band, and the data rates with auto rate fallback are 3, 4.5, 6, 9, 12, 18, 24 and 27 Mbps. For these data rates, the transmitting power is set as 21.5 dBm, and the receiver sensitivity is set as -85, -84, -82, -80, -77, -70, -69 and -67 dBm, respectively. Thus the corresponding effective wireless radio ranges are 384 *m*, 342 *m*, 272 *m*, 216 *m*, 154 *m*, 143 *m*, 60 *m* and 48 *m*, respectively. In addition, we used a 1000 *m* x 1000 *m* area as the simulation field. We used the *UserGraph* model [10] in VanetMobiSim [10] to generate maps for simulation. The area is specified as a grid map in which we set a uniform road section size of 100 *m* and road width of 14 *m* [10]. In each grid, there is a square obstacle. The margin between the road and the obstacle is 5 *m*. That is, the size of an obstacle is 88 *m* x 88 *m*. For city maps of different road section sizes, the road width and the margin are still the same, and the obstacle area is resized accordingly. As revealed in [23], radio transmission is almost blocked by obstacles in a city with densely-populated tall buildings. So in this dissertation we assume the radio is totally blocked by obstacles [13]. In line-of-sight areas, the two-ray ground propagation model [32] was adopted.

As to the mobility impact, we used VanetMobiSim [10] to generate vehicle mobility traces. VanetMobiSim is a vehicle trace generator that includes lane changing, car-following, intersection management and traffic lights models, etc. We adopted the IDM-LC (intelligent

driving model - lane changing) mobility model [10] that has proper speed acceleration and deceleration due to traffic lights and inter-vehicle safety driving behaviors. As shown in Table 3, the settings of city maps and vehicle mobility. The default map size is 1000 *m* x 1000 *m*. We have three map scenarios, map 1 ~ map 3. The number of traffic lights is in proportion to the number of road intersections in each map.

Table 3. Settings of city maps and vehicle mobility.

Settings of city maps [10]	
Field size	Default: 1000 *m* x 1000 *m*
Map layout	*UserGraph* [10] as a grid map with uniform road section size
Road section size	100 *m*
Default map	1000 *m* x 1000 *m* with road section size of 100 m
Map scenario 1 (map 1)	1200 *m* x 1200 *m* with road section size of 100 *m*
Map scenario 2 (map 2)	1200 *m* x 1200 *m* with road section size of 200 *m*
Map scenario 3 (map 3)	1200 *m* x 1200 *m* with road section size of 400 *m*
Number of lanes	2
Number of traffic lights (locations are random)	20 for the default map
	28 for map 1
	10 for map 2
	3 for map 3
Time interval between traffic light change	10 *s*
Vehicle mobility settings [10]	
Minimal stay	5 *s*
Maximal stay	30 *s*
Total number of vehicles	100
Minimal speed	5 *m/s*
Maximal speed	20 *m/s*

As to the network settings, we adopted the Location Aware Routing (LAR) [9], which is a geographical routing protocol that is suited for GPS-enabled networks such as VANETs, as the underlying routing protocol. The settings of an overlay in VANETs are listed in Table 4. The playout buffer time for live streaming is set as 3 seconds [28]; a packet is ignored if it travels from source to receiver exceeding the playout buffer time. We used the constant bit rate (CBR) to generate 64 *kbps*, 128 *kbps* and 256 *kbps* streams, and the packet size is set to 1024 bytes. The source node is randomly chosen.

Table 4. The settings of an overlay in VANETs.

Parameter	**Value**
Simulation time	900 *s*
Underlying routing protocol	LAR [1][9]
Total number of nodes	100
Overlay size (number of overlay nodes)	5, 10 (default), 15, 20
Playout buffer time	3 *s* [28]
Stream rate	64 *kbps*, 128 *kbps*, 256 *kbps*
Packet size (size of a video segment)	1024 bytes
Maximum number of children (MAX_CHILDREN)	6 [30]
MST exchange interval	5 *s*
Probe distance threshold (PROBE_RANGE)	500 *m*
PROBE interval	10 *s*
Maximum number of nodes to probe (MAX_PROBE)	4

Three metrics are defined for evaluation:

(1) *Packet loss rate*: the ratio of the number of data packets lost for an overlay node to the number of data packets generated for a data stream at the source node. The data packets not arrived within playout buffer time are also regarded as packets lost. *Packet loss rate* is to

indicate the fraction of data packets lost in a data stream.

(2) *End-to-end delay*: the elapsed time of non-duplicate data packets delivered from the source node to an overlay node. Note that data packets not arrived within playout buffer time are also included in the calculation of end-to-end delay. *End-to-end delay* is used to derive a proper playout buffer time.

(3) *Control overhead*: the ratio of control bytes to the non-duplicate data bytes received within playout buffer time for an overlay node. *Control overhead* is to indicate the additional cost to receive stream data.

We evaluate the above three metrics for different approaches, under overlay sizes from 5 to 20 nodes. In the simulation results, the following labels are used to represent different cases of overlays, as shown in Table 5. For example, d0b1_1p represents an obstacle-prone static tree overlay. Note that the labels ALMA_b0 and ALMA_b1 mean the ALMA without and with obstacles simulated, respectively.

Table 5. Labels used to represent different cases of overlays.

Label	Meaning
d0	Static overlay
d1	dynamic overlay
b0	no obstacle
b1	obstacle-prone
p1	tree overlay (one parent)
p2	Mesh overlay (two parents)

3.3.1. Packet loss rate evaluation

For each overlay case (see Table 5), the packet loss rate with respect to the number of overlay nodes is shown in Figure 12. We discuss the simulation results in five aspects:

(1) The impact of obstacles:

In Figure 12, it shows that the existence of obstacles significantly affects the stream's packet loss rate. For example, in Figure 12(a), the packet loss rate of an obstacle-prone tree overlay (d1b1_1p) is as high as 4.84 times compared to that of a no-obstacle tree overlay (d1b0_1p). Note that no-obstacle cases are plotted as dotted lines. Other cases also have significant differences, such as d0b1_2p vs. d0b0_2p and d1b1_1p vs. d1b0_1p, etc. In Figure 12(a) ~ (c), the obstacle-prone overlays could increase 0.04 ~ 10.45 times of the packet loss rate compare to the no-obstacle overlays. The simulation results indicate that obstacle-prone environments, e.g., urban VANETs, suffer from significantly higher packet loss than no-obstacle environments.

(2) The improvement of using dynamic overlays instead of static overlays:

To evaluate the impact of using a dynamic overlay, Figure 12(a) shows that d1b1_1p (a dynamic tree overlay) is better than d0b1_1p (a static tree overlay) and d1b1_2p (a dynamic mesh overlay) is better than d0b1_2p (a static mesh overlay), in terms of packet loss rate. Comparing the packet loss rate of the proposed OMV (d1b1_2p) to that of the static mesh overlay (d0b1_2p) [1], an improvement of 27.1 % is obtained, with group sizes of 5 ~ 20 and stream rates of 64 *kbps* ~ 256 *kbps*, on average. The dynamic overlay significantly improves the static overlay. That is, the dynamic overlay is recommended for urban VANETs. In addition, in Figure 12(a) ~ (c), comparing the packet loss rate of the proposed OMV for tree overlays (d1b1_1p) to ALMA, which is also a dynamic tree overlay, an improvement of 7.1 % is obtained, with group sizes of 5 ~ 20 and stream rates of 64 *kbps* ~ 256 *kbps*, on average. It suggests that the overlay adaptation approach of the proposed OMV is better than that of ALMA. This is because that ALMA's parent selection is based on overlay link delay only. Due to high packet loss in urban VANETs, in ALMA, a node may switch to a new parent with a shorter overlay link delay but a higher packet loss rate.

(3) The improvement of using mesh overlays instead of tree overlays:

To evaluate the impact of using mesh overlays, Figure 12(a) shows that d0b1_2p (a static mesh overlay) is better than d0b1_1p (a static tree overlay) and d1b1_2p (a dynamic mesh overlay) is better than d1b1_1p (a dynamic tree overlay), in terms of packet loss rate. Comparing the packet loss rate of the dynamic mesh overlay (d1b1_2p) to that of the dynamic tree overlay (d1b1_1p), an improvement of 41.75% is obtained, with group sizes of 5 ~ 20 and stream rates of 64 *kbps* ~ 256 *kbps*, on average. The two-parent mesh overlay significantly improves the one-parent tree overlay. That is, the mesh structure overlay is recommended for urban VANETs.

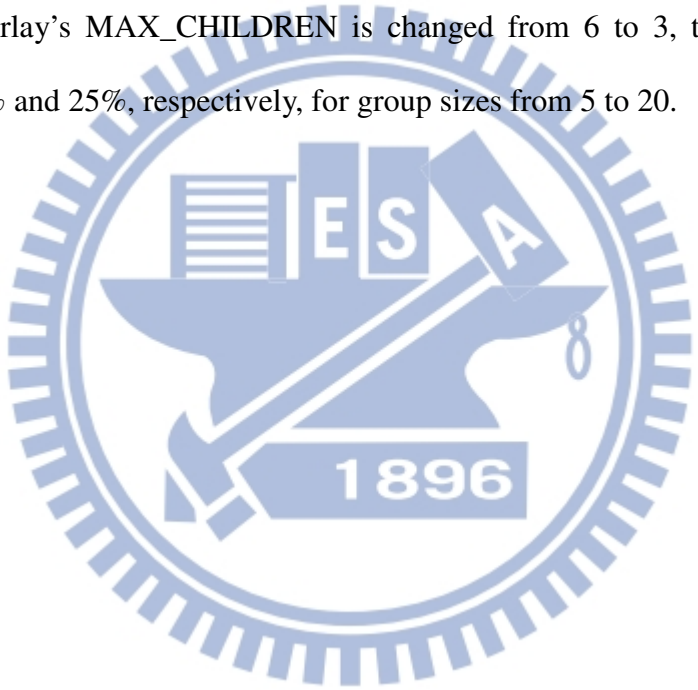
(4) The impact of number of overlay nodes:

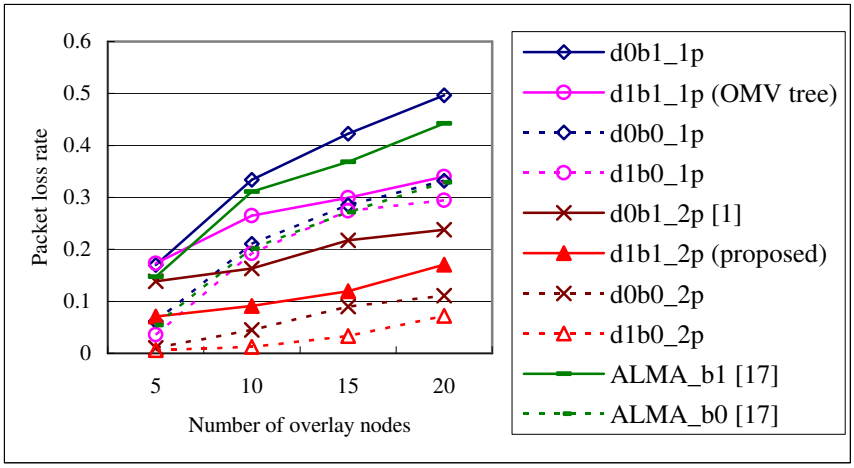
For any case of the overlays in Figure 12(a) ~ (c), the packet loss rate increases as the number of overlay nodes becomes larger. In addition, in Figure 12(a), the packet loss of a tree overlay (d0b1_1p or d1b1_1p) is larger than that of a mesh overlay (d0b1_2p or d1b1_2p), as the number of overlay nodes increases. It suggests that the mesh overlay is more reliable than the tree overlay.

Figure 12(b) and Figure 12(c) use different stream rates, 128 *kbps* and 256 *kbps*, respectively. They show the same observations as Figure 12(a) that dynamic overlays are better than static overlays, and two-parent mesh overlays are better than one-parent tree overlays. Figure 16 shows the impact of a different stream rate to the packet loss. Figure 16(a) is the case of a two-parent mesh overlay (d1b1_2p) and Figure 16(b) is the case of a one-parent tree overlay (d1b1_1p). Both figures show that as the stream rate increases, the packet loss increases. This is due to more data traffic that results in more packets collisions.

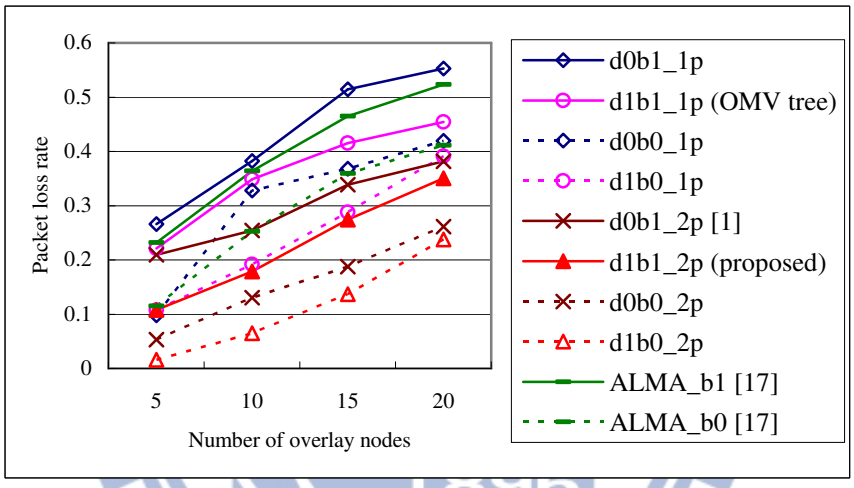
Regarding to the finding that the packet loss rate increases as the number of overlay nodes becomes larger, this is because the increase of overlay size results in a higher tree depth. To validate this, we conducted some experiments to see the impact of a tree depth on performance degradation. The default maximum number of children (MAX_CHILDREN) is 6

in an overlay. If we change MAX_CHILDREN to 3, the depth of the overlay, either a tree or a mesh, would increase. Figure 13(a) and (b) show that an overlay with a higher depth indeed suffers from a higher packet loss rate, under stream rate of 64 and 128 *kbps*, respectively. Note that mesh overlays with different MAX_CHILDREN (d1b1_2p with mc3 and mc6) have smaller differences of packet loss rates than tree overlays with different MAX_CHILDREN (d1b1_1p with mc3 and mc6). This is because that an overlay node disconnection in a tree overlay leads to disconnections of all its descendents, while a mesh overlay is more robust because each overlay node has two parents. For example, as shown in Figure 13(a) and (b), when the tree overlay's MAX_CHILDREN is changed from 6 to 3, the packet loss rates increase about 40% and 25%, respectively, for group sizes from 5 to 20.

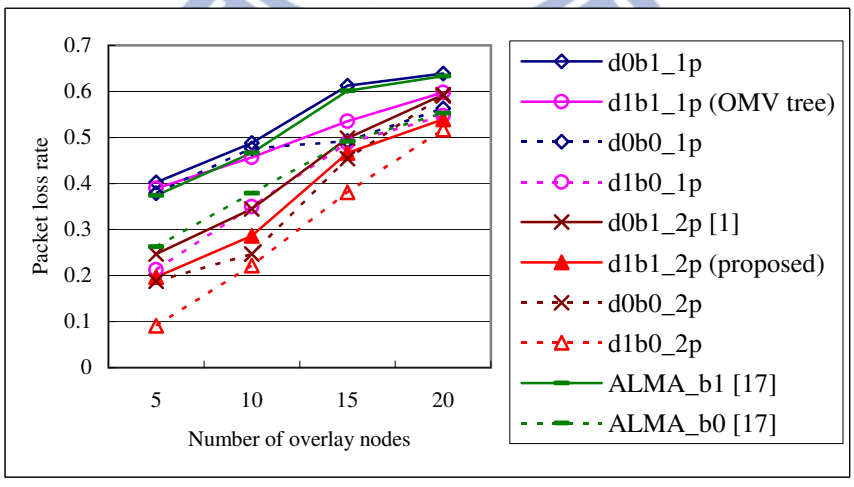




(a) Stream rate of 64 *kbps*.

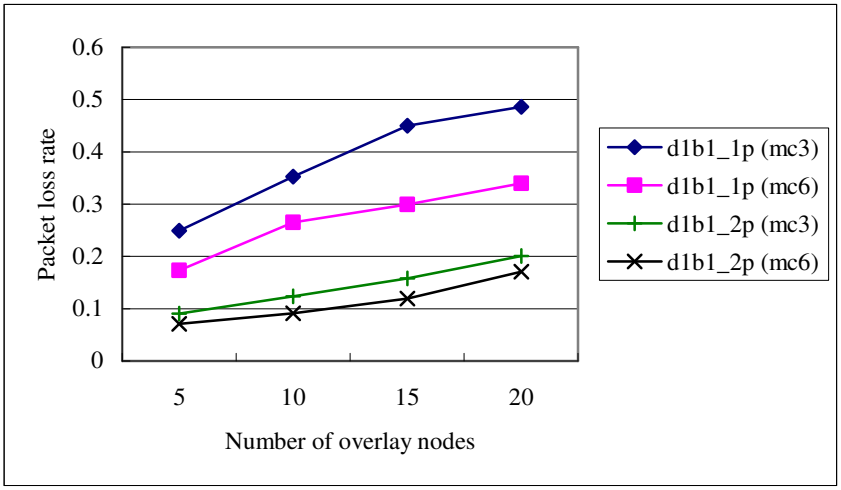


(b) Stream rate of 128 *kbps*.

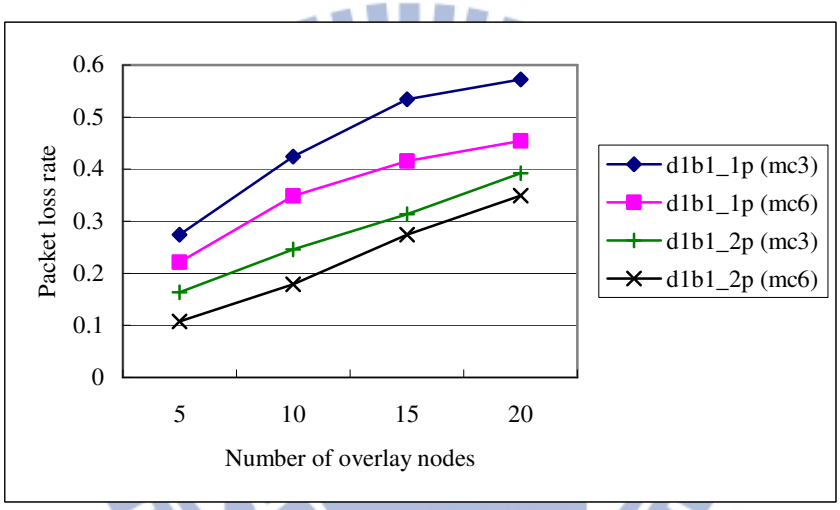


(c) Stream rate of 256 *kbps*.

Figure 12. Packet loss rate comparison among different cases of overlays: (a) stream rate of 64 *kbps*; (b) stream rate of 128 *kbps*; (c) stream rate of 256 *kbps*.



(a) Stream rate of 64 *kbps*



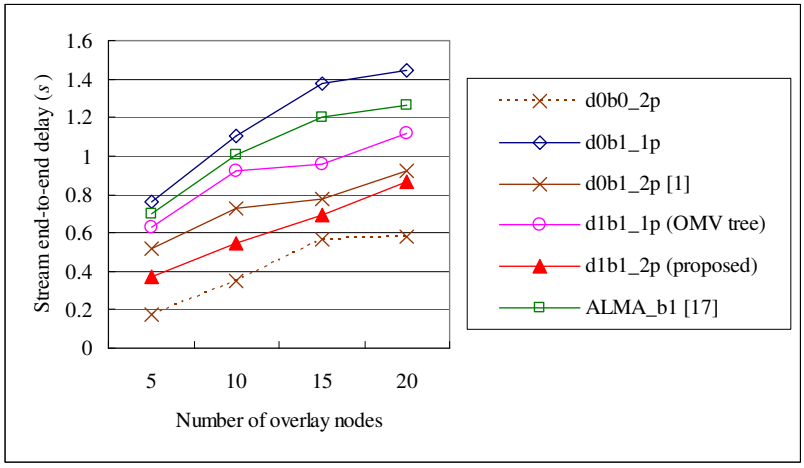
(b) Stream rate of 128 *kbps*

Figure 13. Packet loss rates of the proposed OMV for tree and mesh overlays using different MAX_CHILDREN: (a) stream rate of 64 *kbps*; (b) stream rate of 128 *kbps*.

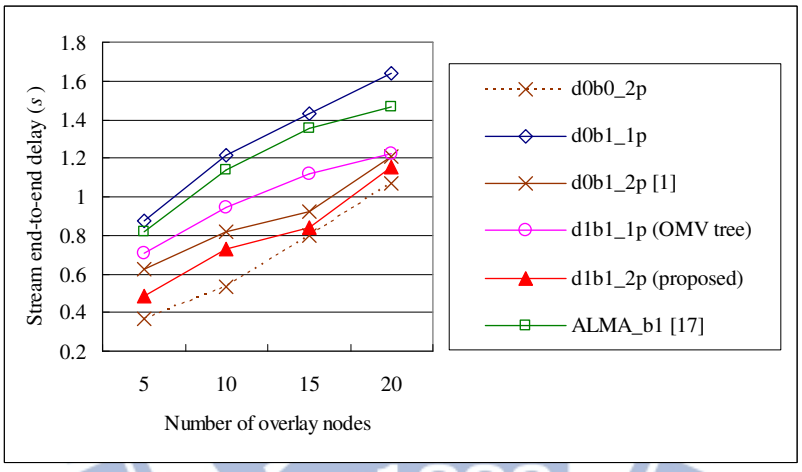
3.3.2. End-to-end delay evaluation

Figure 14 compares the stream's end-to-end delay. Non-obstacle cases are not shown here, since they are not real cases in urban VANETs. Figure 14(a) ~ (c) show that the dynamic overlay has shorter end-to-end delay compared to the static overlay (see the cases of d1b1_1p vs. d0b1_1p and d1b1_2p vs. d0b1_2p). For cases of d0b1_2p vs. d0b1_1p and d1b1_2p vs. d1b1_1p, they show that the mesh overlay has shorter end-to-end delay than the tree overlay.

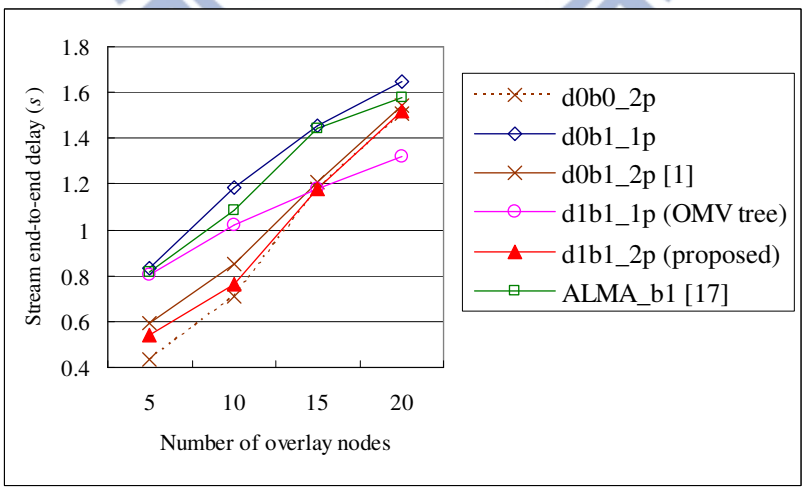
Except for the cases with group sizes of 15 and 20 and stream rate of 256 *kbps*, the end-to-end delay of the dynamic mesh overlay, d1b1_2p, is lower than that of the dynamic tree overlay, d1b1_1p. This is due to that a mesh overlay enhances the overlay connectivity so as to reduce the end-to-end delay. But under a high stream rate, the redundant data of a mesh overlay also grows larger and causes more traffic congestion and delay. For any case in Figure 14(a) ~ (c), the end-to-end delay of OMV is lower than 1.7 *s*. Comparing the end-to-end delay of the OMV (d1b1_2p) to that of the static mesh overlay (d0b1_2p) [1], a decrease of 11.7% is obtained, with group sizes of 5 ~ 20 and stream rates of 64 *kbps* ~ 256 *kbps*, on average. Comparing the end-to-end delay of the proposed OMV for tree overlays (d1b1_1p) to that of ALMA, in Figure 14(a) ~ (c), a decrease of 13.1% is obtained, with group sizes of 5 ~ 20 and stream rates of 64 *kbps* ~ 256 *kbps*, on average. This is because that ALMA chooses a new parent with lower overlay link delay, which seems to lead to shorter end-to-end delay. However, ALMA looks for nodes in a near overlay neighborhood first. Instead, in the proposed OMV for tree overlays, an overlay node may utilize the movement state table (MST) to choose nodes to probe. This allows the OMV to probe nodes close to the overlay node even they are not in its near overlay neighborhood. Therefore, for the OMV, the possibility of finding a new parent that brings a lower packet loss rate as well as a lower end-to-end delay increases.



(a) Stream rate of 64 *kbps*



(b) Stream rate of 128 *kbps*



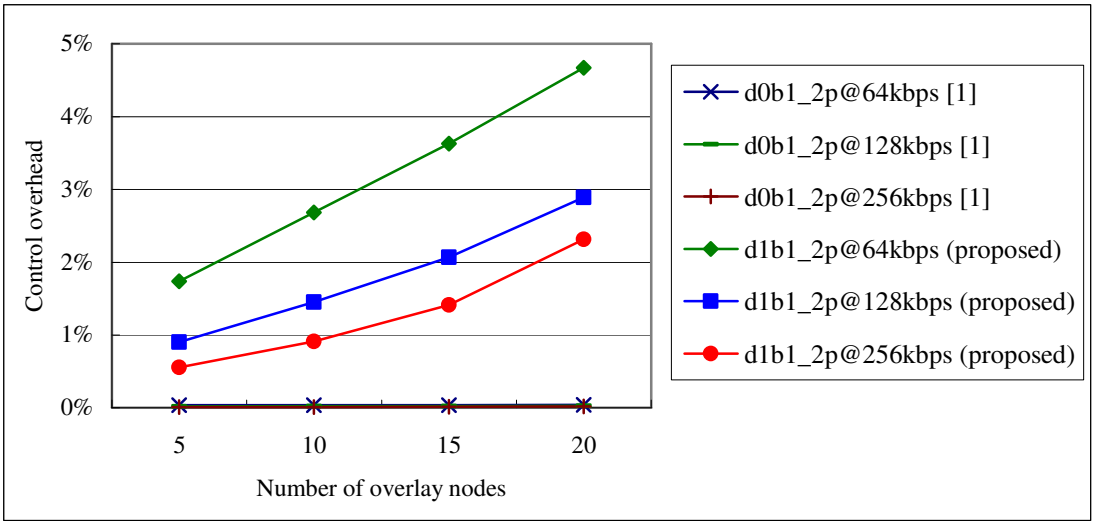
(c) Stream rate of 256 *kbps*

Figure 14. End-to-end delay comparison among different cases of overlays: (a) stream rate of 64 *kbps*; (b) stream rate of 128 *kbps*; (c) stream rate of 256 *kbps*.

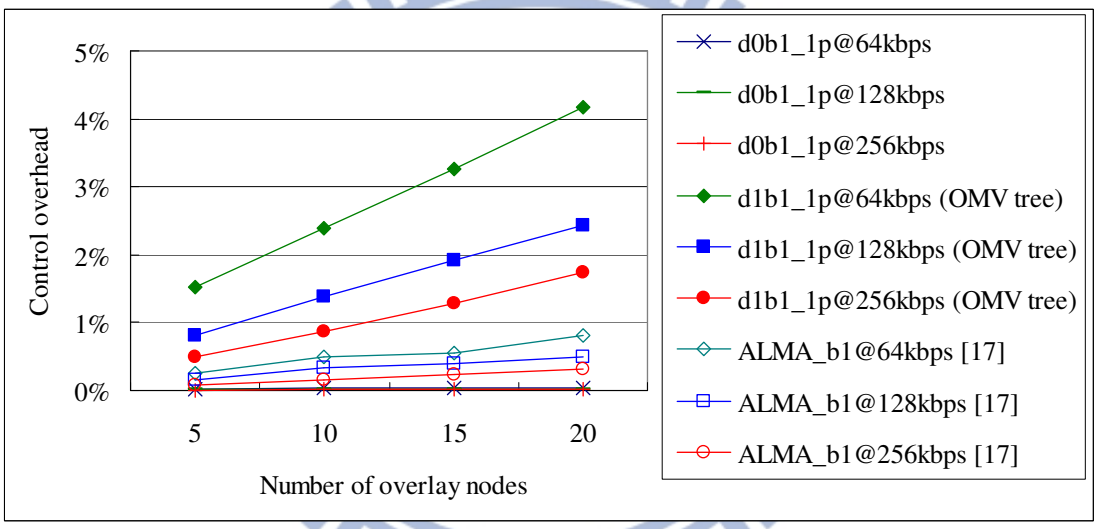
3.3.3. Control overhead evaluation

Control overhead comes from the overlay's construction and dynamic adaptation. Figure 15 shows the additional cost due to applying the proposed QoS-satisfied dynamic overlay adaptation approach to the static overlay. For mesh overlays, as shown in Figure 15(a), static overlays (d0b1_2p cases) have almost no control overhead, because only initial overlay construction incurs small control overhead; dynamic overlays (d1b1_2p cases, the proposed OMV) use periodic PROBE procedure and thus incur more control overhead. The proposed OMV incurs control overhead from 0.55% (64 *kbps* with 5 overlay nodes) to 4.67% (256 *kbps* with 20 overlay nodes). On average, the proposed OMV incurs control overhead of only 2.1%, with group sizes of 5 ~ 20 nodes and stream rates of 64 *kbps* ~ 256 *kbps*. As to tree overlays, as shown in Figure 15(b), the results are similar to those of mesh overlays in Figure 15(a). Note that the control overhead of ALMA is lower than that of the proposed OMV for tree overlays. This is because that in the OMV each overlay node needs to exchange MSTs (movement state tables) with its parent and children periodically. Finally, in Figure 15(a) and (b), the control overhead for an overlay with a high stream rate is lower than that with a low stream rate. This is because the incurred control bytes remain the same regardless of the stream rate.

Note that in Figure 15, control overhead increases linearly with size of the group; however, the group size will be bounded for feasible multimedia streaming. As we will see in Table 6, it shows a feasible combination of road section size, group size, and stream rate. In Table 6, under a total of 100 nodes, a group size within 20 is suggested for feasible multimedia streaming in urban VANETs, and the corresponding control overhead is less than 5%, as shown in Figure 15. If the group size needs to be increased, the total number of nodes needs to be increased as well in order to achieve feasible multimedia streaming.



(a) Mesh overlays



(b) Tree overlays

Figure 15. Control overhead comparison among dynamic and static overlays: (a) mesh overlay; (b) tree overlay.

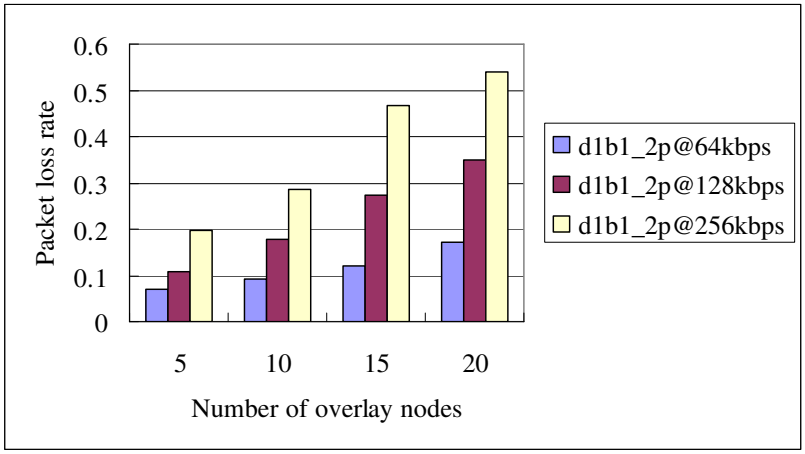
3.3.4. Feasibility evaluation of the proposed OMV

(1) Video streaming feasibility study

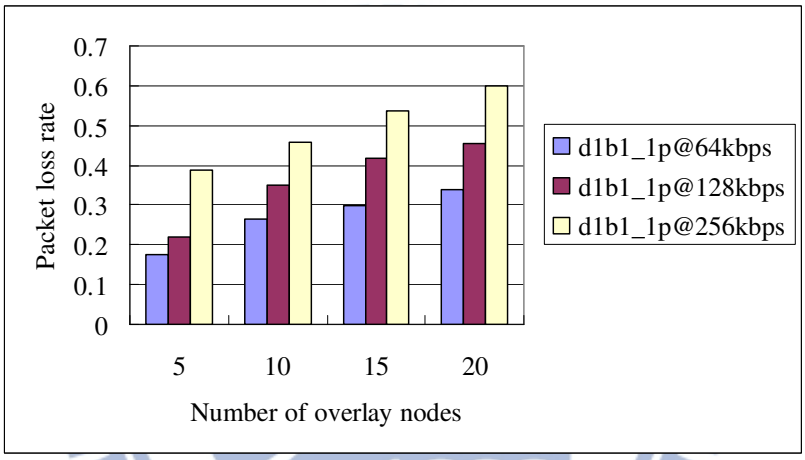
To understand the resulting video quality, in the following, we discuss the relationship among PSNR (peak-signal-to-noise ratio) [29], MOS (mean opinion score) [26], and packet

loss rate. PSNR is a metric to indicate the quality of a received video. MOS is a score of human observers' opinions to a video's quality. With advanced error-resilient video coding techniques developed, a video could be recovered well at the receivers, and the resulting visual quality is good [33][34][35][36]. For example, some recent error-resilient video coding techniques were reported to be able to tolerate a packet loss rate of 30% [33][34][35][36]. That is, if an advanced error-resilient video coding technique is applied to a video, under a packet loss rate of 30%, the video's PSNR can still remain above 25 [33][34][35][36]. PSNR of 25 ~ 31 falls in the MOS range of 'fair' quality [27][40][41]. If the MOS range of 'good' ('excellent') is required, a packet loss rate of less than 15% (8%) is required [33][34][35][36]. Therefore, based on the above discussion, we set 0.3 as a threshold of the packet loss rate to obtain fair video quality. For each overlay case, we will point out under what configurations to obtain fair video quality. As shown in Figure 16(a), for the two-parent mesh overlay, the packet loss rate is within 0.2 for a 64 *kbps* stream with 5 to 20 overlay nodes, and so do the cases for a 128 *kbps* stream with 5 to 10 overlay nodes; for a 256 *kbps* stream with 5 to 10 overlay nodes, the packet loss rate is within 0.3. As shown in Figure 16(b), for the one-parent tree overlay, the packet loss rate is within 0.3 for a 64 *kbps* stream with no more than 15 overlay nodes, and so does for a 128 *kbps* stream with 5 overlay nodes.

In summary, a two-parent dynamic overlay is feasible for video streaming under the condition of (1) a low stream rate of 64 *kbps*, or (2) a medium stream rate of 128 *kbps* with a medium overlay size of 15 overlay nodes, and (3) a high stream rate of 256 *kbps* with a small overlay size of 10 overlay nodes. A one-parent dynamic tree overlay is feasible for video streaming under the condition of (1) a low stream rate of 64 *kbps* with a medium overlay size of 15 overlay nodes, or (2) a medium stream rate of 128 *kbps* with a small overlay size of 5 overlay nodes.



(a) Mesh overlays



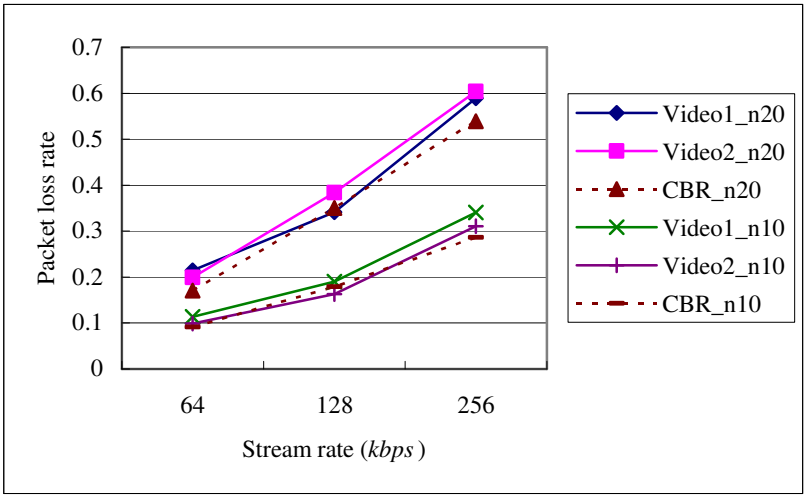
(b) Tree overlays

Figure 16. The impact of a different stream rate to the packet loss rate: (a) dynamic two-parent mesh overlay; (b) dynamic one-parent tree overlay.

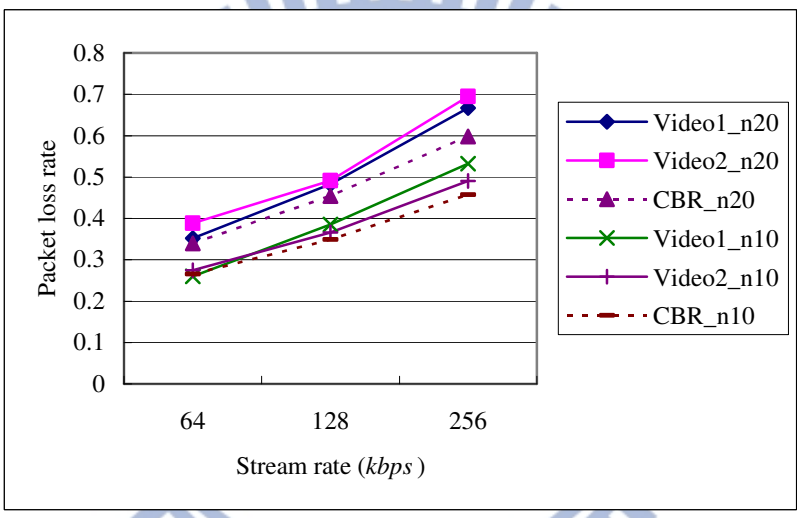
(2) Feasibility test using two real video clips with vehicular scenarios

We have also tested our scheme using two real video clips with vehicular scenarios, *Carphone* and *Highway* [39], and experimental results are shown in Figure 17 and Figure 18. The two real video clips are YUV video sequences in QCIF format (video resolution of 176 x 144 pixels/frame). *Carphone* shows a man talking in a car. *Highway* shows the front view of driving over a highway. We retrieved frames from the YUV video sequences for frame rate of 15 Hz, i.e. 15 frames/sec. The length of *Highway* with 1000 frames is 66.6 seconds. Since the

length of *Carphone* with 150 frames is only 10 seconds, we replay *Carphone* up to 66.6 seconds. We used FFmpeg [37] and MP4Box [38] to compress both YUV video sequences in H.264/MPEG4 standard with different bit rates, such that the compressed files would result in 64 *kbps*, 128 *kbps* and 256 *kbps* data flows. We evaluated the packet loss rates of the two video clips, using group sizes of 10 and 20. In Figure 17 and Figure 18, *Carphone* is labeled as Video1, *Highway* is labeled as Video2, and CBR is also shown here for references. The group size is also included in a label. Figure 17(a) shows that for the proposed OMV for mesh overlays, under the same stream rate and the same group size, the packet loss rates of the two video clips and CBR are close. This means that the performance of the proposed OMV is feasible for various real video clips. Similarly, for the proposed OMV for tree overlays, as shown in Figure 17(b), the packet loss rates of the two video clips and CBR are also close. In addition, Figure 18(a) and (b) show that for the proposed OMV for mesh overlays and the proposed OMV for tree overlays, respectively, under the same stream rate and the same group size, the end-to-end delays of the two video clips and CBR are close. In summary, Figure 17 and Figure 18 show the feasibility of the proposed OMV for real video.

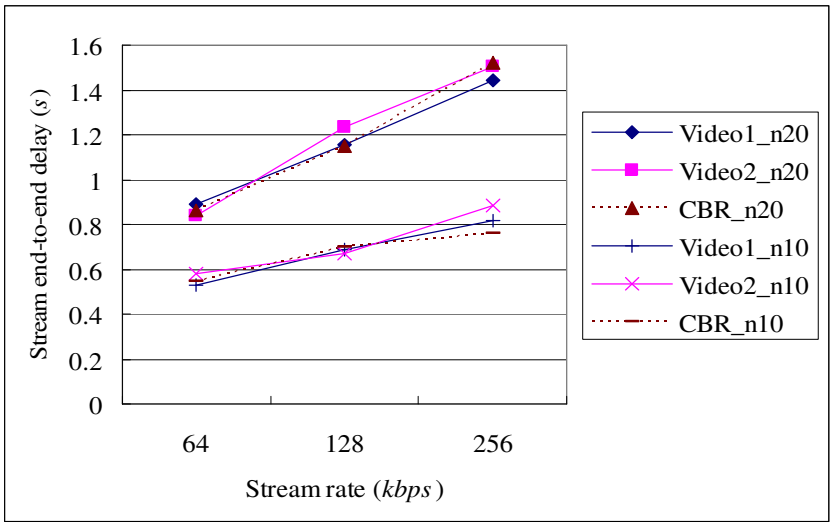


(a) Mesh overlays

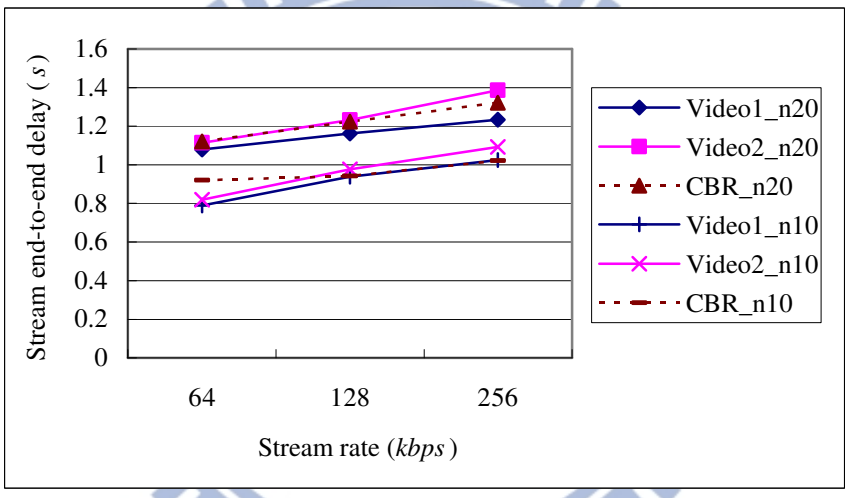


(b) Tree overlays

Figure 17. Packet loss rate comparison among different video clips: (a) dynamic two-parent mesh overlays; (b) dynamic one-parent tree overlays.



(a) Mesh overlays



(b) Tree overlays

Figure 18. End-to-end delay comparison among different video clips: (a) dynamic two-parent mesh overlays; (b) dynamic one-parent tree overlays.

3.3.5. The impact of road section sizes to overlays performance

We want to investigate the impact of road section sizes to overlays performance, with various stream rates and overlay sizes. In Figure 19 and Figure 20, the city map sizes are set as 1200 *m* x 1200 *m*, instead of 1000 *m* x 1000 *m*, so as to have road section sizes of 100 *m*, 200 *m*, and 400 *m*. The other settings remain the same.

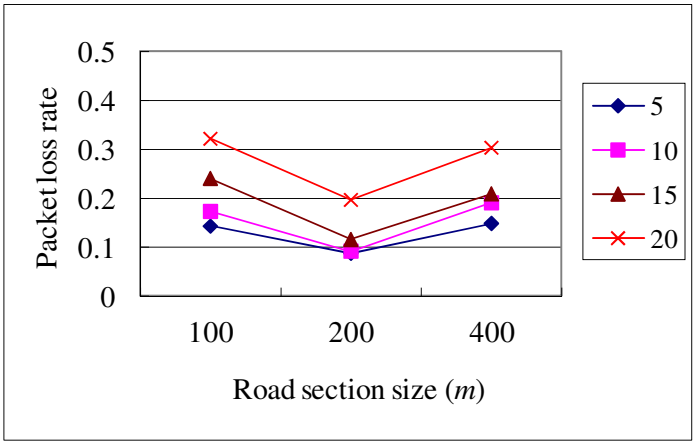
(1) The impact of road section sizes to packet loss rates:

Figure 19 shows the packet loss rate with road section size of 200 *m* is lower than that of road section sizes 100 *m* and 400 *m*, respectively. With the same number of nodes, when the road section size increases, the number of nodes on each road section becomes larger, which implies higher connectivity, and thus it is good for packet routing. However, a larger road section size results in fewer road intersections in the city map. As a result, there are fewer choices for nodes to route packets from one road section to another road section, which is a disadvantage to packet routing. Besides, higher connectivity among nodes also brings more packet collisions and thus results in more packet loss. Therefore, under these positive and negative impacts, a road section size of 200 *m* is more preferred to the others, as shown in Figure 19. These findings suggest us that based on a road section size and number of overlay nodes, we may choose an appropriate stream rate to have a fair video quality with a packet loss rate within 0.3, according to Figure 19.

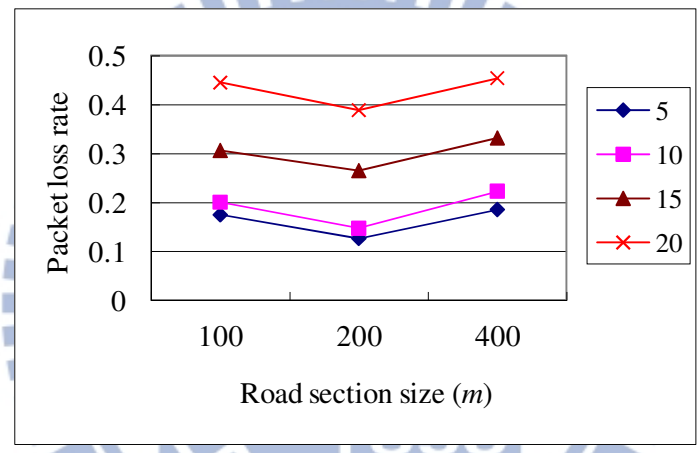
(2) The impact of road section sizes to end-to-end delays:

Figure 20 shows the stream's end-to-end delays under different road section sizes. The end-to-end delay of road section size 200 *m* is the lowest. The reason is similar to that in Figure 19. When the road section size grows from 100 *m* to 200 *m*, the nodes densities become denser in road sections, which are good for packet routing, so that the end-to-end delay decreases. However, when the road section size grows from 200 *m* to 400 *m*, the end-to-end delay will increase. This is because that a larger road section size results in fewer choices for packet routing among different road sections. However, for the cases of overlay of size of 15 and 20 in Figure 20(b) and (c), when the road section size increases from 200 *m* to 400 *m*, their end-to-end delays decreases. This is because that under a large overlay size, a high stream rate and a large road section size, packet loss becomes more serious. As a result, most of the received packets have short delay because the packets with long delay may be lost

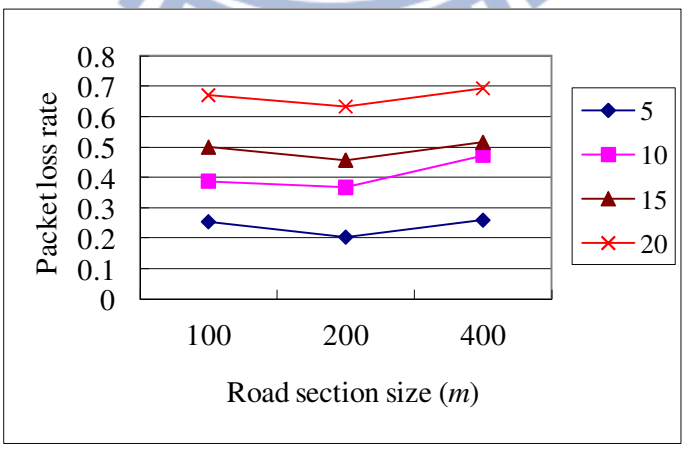
before being received.



(a) Stream rate of 64 *kbps*

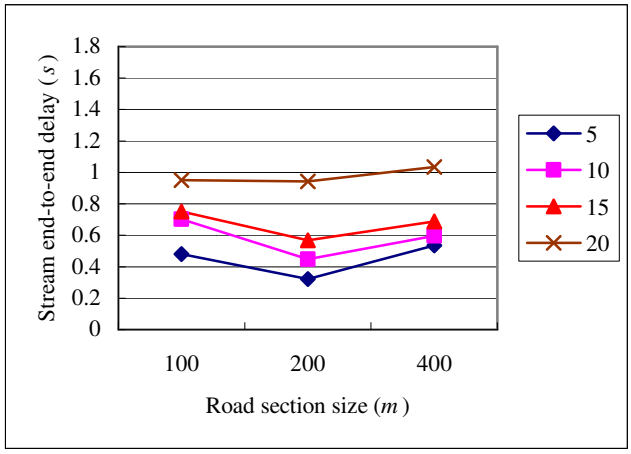


(b) Stream rate of 128 *kbps*

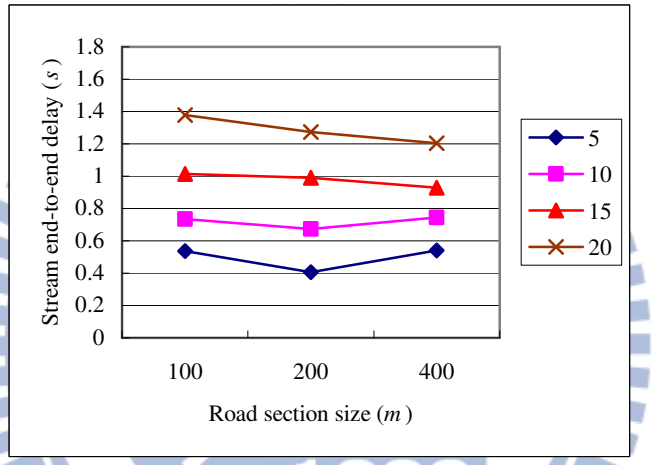


(c) Stream rate of 256 *kbps*.

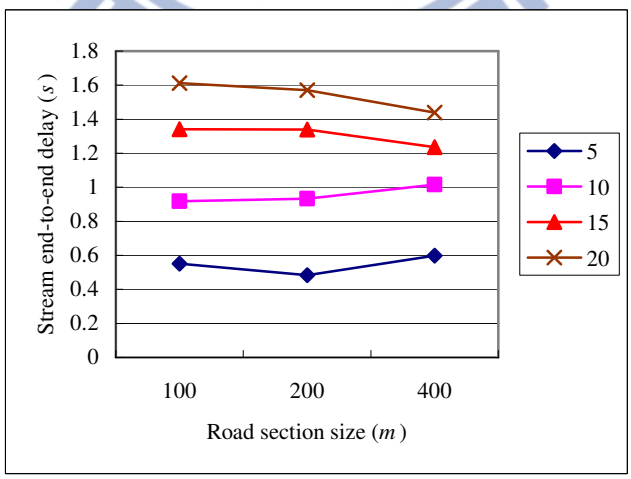
Figure 19. Packet loss rates under different road section sizes: (a) stream rate of 64 *kbps*; (b) stream rate of 128 *kbps*; (c) stream rate of 256 *kbps*.



(a) Stream rate of 64 *kbps*



(b) Stream rate of 128 *kbps*



(c) Stream rate of 256 *kbps*

Figure 20. End-to-end delays under different road section sizes: (a) stream rate of 64 *kbps*; (b) stream rate of 128 *kbps*; (c) stream rate of 256 *kbps*.

(3) Video streaming feasibility study

Remind that we used 0.3 as a threshold of the packet loss rate in order to have fair video quality. In Figure 19(a), a stream rate of 64 *kbps* is feasible for any number of overlay nodes from 5 to 20 combined with all road section sizes of 100 *m*, 200 *m* and 400 *m*, except for overlay nodes number of 20 and road section size of 100 *m*. In Figure 19(b), a stream rate of 128 *kbps* is feasible for any number of overlay nodes from 5 to 15 combined with all road section sizes of 100 *m*, 200 *m* and 400 *m*, except for overlay nodes number of 15 and road section size of 400 *m*. In Figure 19(c), a stream rate of 256 *kbps* is only feasible for overlay nodes number of 5 combined with all section sizes of 100 *m*, 200 *m* and 400 *m*. Finally, in Table 6 we summary feasible stream rates for different road section sizes and overlay sizes. For example, for a group size of 15 and a road section size of 200 *m*, a feasible stream rate is at most 128 *kbps* in urban VANETs.

Table 6. Feasible stream rates for different road section sizes and overlay sizes.

Road section size	**Overlay size** (Total number of nodes: 100 Field size: 1200 *m* x 1200 *m*)			
	5	10	15	20
100 *m*	≤256 *kbps*	≤128 *kbps*	≤128 *kbps*	NA
200 *m*	≤256 *kbps*	≤128 *kbps*	≤128 *kbps*	≤64 *kbps*
400 *m*	≤256 *kbps*	≤128 *kbps*	≤64 *kbps*	≤64 *kbps*

3.3.6. Discussion

A heuristic used in the proposed PROBE procedure for VANETs, optimized parent selection, had also been used in MANETs. However, in the proposed OMV for VANETs, the parent selection is based on both packet loss rate and end-to-end delay, which is more effective than that used by other approaches for MANETs. In low mobility environments, such as MANETs, the effect of using packet loss rate to determine a better parent is limited.

This is because that if nodes are dense enough and their mobility is low, the network connectivity is high. However, in high mobility environments, such as VANETs, broken links occur frequently. In addition, there are obstacles, such as buildings, in urban VANETs that may result in broken links. Especially, in an overlay, an overlay link connecting two overlay nodes may actually consist of multiple physical links. As a result, the packet loss rates of nodes at different depths in a tree could be much different. Therefore, in high-mobility and obstacle-prone environments, such as urban VANETs, using the packet loss rate to determine a better parent is useful. In addition, the nature of high packet loss in urban VANETs may dramatically degrade an overlay's performance. The proposed OMV targets at this situation by using packet loss rate and end-to-end delay for group nodes to switch to new parents that can offer better QoS. Furthermore, compared to other methods used in VANETs that need a high or medium density of group nodes, as shown in Table 1, the proposed OMV can get the help of non-member nodes to assist forwarding data packets via the overlay. In this respect, the proposed OMV is more effective than other methods used in VANETs in terms of reducing packet loss.

In the proposed PROBE procedure that involves optimized parent selection, instead of only considering the delay between a probed parent and the child, we also consider the end-to-end delay, from the source to a probed parent to the child. The advantage of using end-to-end delay is to ensure that live multimedia frames can be played in time. Finally, note that the OMV's limitation is that in low mobility and no-obstacle environments, the performance of the proposed OMV is only comparable to that of other overlay multicast approaches.

Chapter 4

Proposed road-based multipath routing protocol for urban VANETs

In this chapter, we propose a *road-based multipath routing protocol for urban VANETs* (RMRV), which can find multipath paths according to the road map. RMRV estimates paths' future life periods so as to dynamically switch to use another path before the current path disconnects.

4.1. Problem description

In road-based routing, a path is composed of a succession of road sections (RSs). As shown in Figure 21, for example, path 1 is composed of RSs [0, 1, 2, 5, 8]. Besides, multiple paths, such as paths 1 ~ 3, can be found between nodes S and D. However, due to node mobility, an RS cannot always remain connected, and thus the associated path it belongs to also disconnects. For example, if RS 2 is disconnected, the associated path 1 is also disconnected. Therefore, we propose RMRV to resolve the following three issues:

(1) How to find multiple paths between source and destination?

(2) A path's life periods can be estimated from the associated RSs' life periods. The duration of an RS being connected is termed as a life period. As time elapses, a connected RS may become disconnected and later connected again due to nodes mobility. An RS's life periods depend on its associated nodes' positions and speeds. Therefore, according to the nodes mobility in each RS of a path, we can predict the path's life periods in the near future.

(3) With the multiple paths found by (1) and the paths' future life periods estimated by (2), the source node can dynamically switch a connected path to send data packets in anticipation of possible current path failures.

4.2. Multipath discovery

A source node initiates route discovery if it has data packets to send to the destination node but no path available. A route request packet (RREQ) is generated at the source node, and the RREQ packet is broadcasted to the network so as to eventually reach the destination node. The RREQ packet will record the RSs it has visited in an RS list. The RS list in the RREQ packet is updated when the packet enters a new RS. To avoid broadcast storm, for a node receiving an RREQ packet, the node checks whether it has received the same RREQ packet. If not, the node re-broadcasts the RREQ packet. To check whether the same RREQ packet has been received, each node needs to maintain an RREQ table. In traditional routing protocols, the RREQ table records a received RREQ packet's source address, destination address and a unique sequence number generated at the source node. In the proposed RMRV, we additionally save the RS list in the received RREQ packets to the RREQ table. For example, if a received RREQ packet p has the same source address, destination address, sequence number and RS list with entry e in the RREQ table, then we discard this RREQ packet. In addition, we give an upper bound of overlapping RSs, o_u. If packet p's and entry e's RS lists are not the same, but they have more than o_u overlapping RSs, then packet p is also discarded. Note that the beginning and ending RSs, for example, RS 0 and RS8 in Figure 21, are not counted into o_u, since they must be overlapped among paths. The purpose of setting an o_u is to avoid generating too many similar paths that may fail at the same time.

Besides, if a node receives an RREQ packet that includes the RS number of the RS that the node currently resides in, to avoid the loop problem, the node just drops the RREQ packet.

For example, if the node currently resides in RS 3 and it receives an RREQ packet with RS list of [5, 3, 2, 7, 8], it drops the RREQ packet to avoid the loop problem.

Finally, the destination node may receive multiple RREQ packets with different RS lists. Each of these RS lists is a path from source to destination. For each path, the destination node responds the source node with a route reply (RREP) packet. The RS list of each path is contained in the RREP packet, and the RREP packet will be relayed to the source node by geographical forwarding according to the RS list of the path. As the RREP packet being relayed along the RSs, the life periods of each RS are estimated, which will be used to predict the future connectivity of each path.

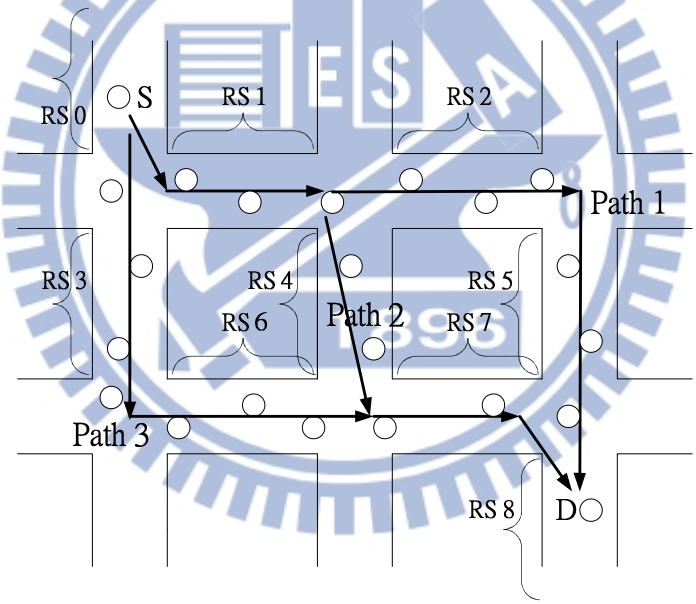


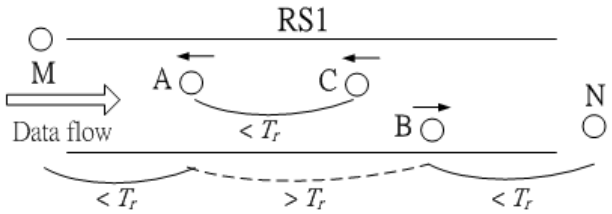
Figure 21. Road-based multipath routing: three paths from the S to D are discovered.

4.3. RS life periods estimation

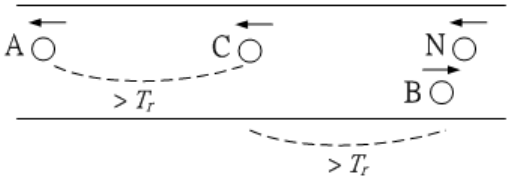
4.3.1. RS connectivity problem

In the proposed RMRV, geographical forwarding is used to relay packets through RSs. Take road section RS1 in Figure 22(a) as an example. Suppose that packets are now relayed to

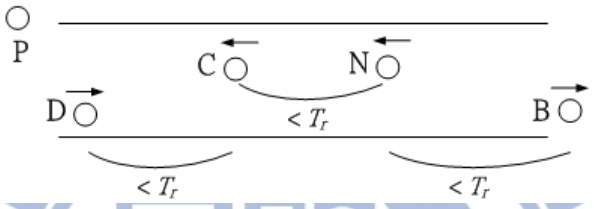
RS1 from node M, received by node A. Node A is currently located at the left hand side of RS1. Then node A would relay packets to its farthest neighbor node, node C, toward the other end of RS1. Then node C can relay to node B, and node B can relay to node N. Therefore, RS1 is connected because packets may pass RS1 with nodes A→C→B, and RS1 is also connected with its adjacent RSs with nodes M→A and B→N. In this way, as long as each RS in a path remains connected, and each RS also remains connected with its adjacent RSs, then packets can successfully be relayed from source to destination. However, the connectivity of an RS will change due to node mobility. For example, in Figure 22(a), assume the current time is t_0 and all nodes will keep moving with their current speed or may make turns at road intersections, and some nodes in other RSs will move into RS1 if their turn signals indicate their future moving directions to RS1. Then, we may estimate the connectivity of RS1 in the near future, as follows. At time $t_0 + 2$ (2 seconds after t_0), as shown in Figure 22(b), the distance between nodes A and C exceeds radio transmission range T_r, and thus the path in RS1 disconnects. Then, at time $t_0 + 4$, as shown in Figure 22(c), node A leaves RS1 and node D enters RS1, and node N can relay packets for nodes C and B. Thus, RS1 becomes connected again. Later, at time $t_0 + 6$, as shown in Figure 22(d), RS1 becomes disconnected due to the broke link between nodes N and B. In summary, from t_0 to $t_0 + 6$, RS1's life periods are $[t_0, t_0 + 2]$ and $[t_0 + 4, t_0 + 6]$. To deal with a general case of an arbitrary number of nodes, we propose a *space-time planar graph* approach to model and resolve the RS connectivity problem, as described in the following.



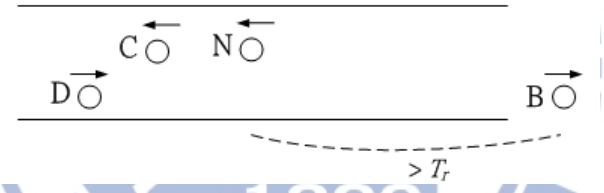
(a) t_0



(b) $t_0 + 2$



(c) $t_0 + 4$



(d) $t_0 + 6$

Figure 22. An example of the RS connectivity problem at each time instance.

4.3.2. Proposed space-time planar graph approach for predicting an RS's connectivity

A local map with RS C_1-C_2 is shown in Figure 23(a). C_1 and C_2 and are road intersections. We divide RS C_1-C_2 into multiple slots, X, Y, Z and W. The leftmost and rightmost slots, X and W, are the areas that the radio coverage of the nodes in other RSs adjacent to RS C_1-C_2 can reach. Then, the remaining part of RS C_1-C_2 is further divided into slots of length T_r/2, for instance, slots Y and Z. T_r is the radio transmission range. In this way, as long as there is at least one node in each slot of RS C_1-C_2, then RS C_1-C_2 is connected. However, nodes are

mobile. The arrows in Figure 23(a) represent nodes' moving directions. Currently, RS C_1-C_2 is connected, but how long will it remain connected? In other words, our goal is to derive the time periods that each slot contains at least one node.

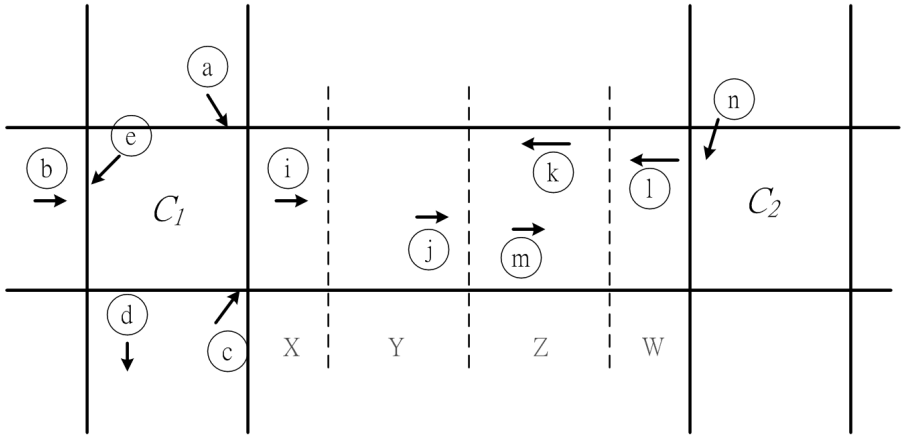
Firstly, we describe how mobility data of all nodes in an RS are shared. Mobility data includes a node's coordinate, speed and moving direction. As mandated by DSRC [56][57], every 300 *ms* each vehicle sends a beacon message with a range of about 200 - 300 m [58]. Thus the mobility data can be appended to the beacon message. In addition, each node maintains a neighbor table, which is updated upon receiving a beacon message. Furthermore, the mobility data of the nodes located in adjacent RSs, such as nodes a, b and c in Figure 23(a), which are node *i*'s neighbors, can also be obtained by node *i*. And so does node *n*'s mobility data can be obtained by node l. In Figure 23(a), if a route reply (RREP) packet is propagated from C_2 to C_1, for each node that relays the RREP packet, the relay node's neighbor table is piggybacked to the RREP packet. When the last node in the RS, for example, node i in Figure 23(a), receives the RREP packet, node i is in charge of estimating the future connectivity of RS C_1-C_2, using the proposed space-time planar graph approach, as follows.

Figure 23(b) shows the space-time planar graph of Figure 23(a). This planar graph is generated by node i, upon node i receiving an RREP packet. In this graph, each node is represented as an arrow. The arrow shows a node's position within RS C_1-C_2 along time. That is, a point (s, t) on an arrow p represents that node p is located at position s of RS C_1-C_2 at time t. The starting time of x-axis in this planar graph is t_0, which is the moment that node i receives an RREP packet. With a node's current position and speed, which can be retrieved from the mobility data piggybacked in the RREP packet, the node's trace (i.e. the arrow of the node in the space-time planar graph) can be easily generated as follows. The slope of the arrow is the node's speed. At time t_0, if the node is currently located at position s in RS C_1-C_2, the starting point of the arrow is on the C_1-C_2 axis's position s. As to the nodes not currently

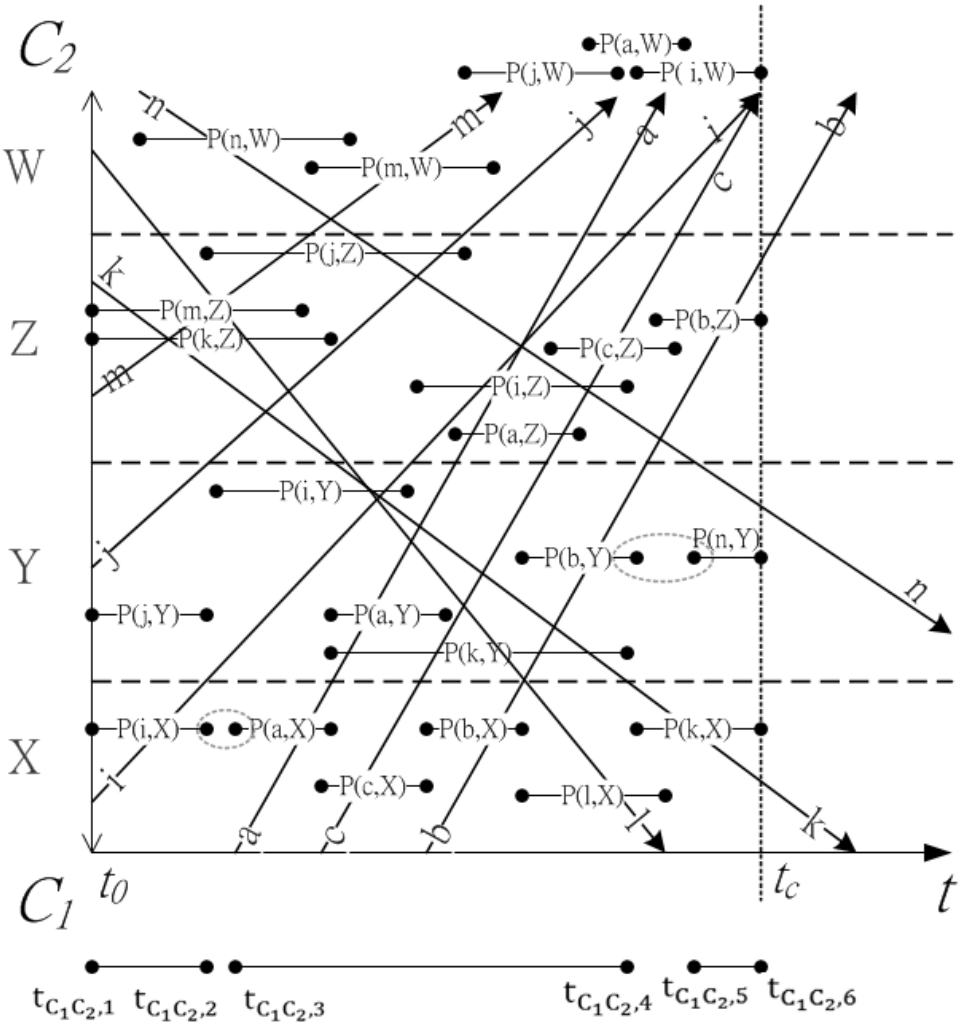
located in RS C_1-C_2, but will move to RS C_1-C_2 based on their turn signals, we call these nodes as entering nodes. In the space-time planar graph, for the starting point (s_n, t_n) of an entering node's arrow, s_n must be C_1 or C_2, and t_n can be derived according to the remaining time of the red light from the city's public ITS (Intelligent Transportation System). When an ITS is not available, t_n can be estimated by the entering node's current speed and its distance to C_1 (or C_2). Finally, we assume the average node speed in the RS as the slope of an entering node's arrow.

Based on the generated space-time planar graph, we can predict the connectivity of RS C_1-C_2 from time t_0 to t_c, as shown in Figure 23(b). Take node m as an example. Node m moves from slot Z to slot W. It stays within slot Z for a period of time P(m, Z), and stays within slot W for P(m, W). P(m, Z) can be obtained by projecting the line segment of arrow *m* in slot Z to the *t* axis. After that, we can examine whether all the projections within slot Z can cover the time from t_0 to t_c. As shown in Figure 23(b), in slot Y, there is a gap between P(b, Y) and P(n, Y). That is, during this period of time, slot Y has no node within it. Similarly, we can deal with other slots in the same way. There is another gap in slot X, between P(i, X) and P(a, X). Recall that an RS is connected if each of its slots has at least one node. By unioning these gaps as the time periods that RS C_1-C_2 is disconnected, we can obtain the time periods that RS C_1-C_2 is connected: $[t_{C_1C_2,1}, t_{C_1C_2,2}]$, $[t_{C_1C_2,3}, t_{C_1C_2,4}]$ and $[t_{C_1C_2,5}, t_{C_1C_2,6}]$. We call them the life periods of RS C_1-C_2 in $[t_0, t_c]$.

The proposed space-time planar graph is a lightweight algorithm. Let *s* denotes the number of slots in an RS and *n* denotes the number of vehicles in the RS, then the computation complexity of the space-time planar graph for the RS would be $O(n) + O(sn) + O(sn)$. The $O(n)$ is the cost for generating the arrows of the *n* nodes, the first $O(sn)$ is the cost for deriving the *n* arrows' projections in each of the *s* slots, and the second $O(sn)$ is the cost for intersecting the *sn* arrow projections.



(a)



(b)

Figure 23. An example of using the space-time planar graph approach to estimate future connectivity of an RS. (a) Node distribution in RS C_1-C_2. (b) Space-time planar graph of RS C_1-C_2.

4.4. Path life periods estimation

Using the space-time planar graph, we can obtain every RS's life periods. Then, a path's life periods can be obtained by the intersection of its RSs' life periods. When the node in charge of processing the Space-time planar graph receives an RREP packet, the RS life periods generated in the previous RS can be extracted from the RREP packet. We intersect the extracted data with the current RS's to obtain the latest result. In this way, all the previous RSs' life periods can be integrated into one copy. For example, if the previous RS's integrated life periods are [1s, 2s] and [3s, 4s], and the current RS's life periods are [1.5s, 2.5s] and [2.8s, 3.8s], then the new integrated life periods are [1.5s, 2s] and [3s, 3.8s]. Then, only this node's (the node in charge of processing the space-time planar graph) neighbor table and the new integrated life periods are attached to the RREP packet. The RREP packet is then relayed to the next RS. Finally, when the source node receives the RREP packet, the estimated life periods of the whole path is obtained. Note that packet transmission speed is much higher than node moving speed, thus the RREP packet can be sent back to the source node successfully before potential path disconnections. In addition, to efficiently control the volume of RREP packets, the neighbor table piggybacked in an RREP packet is removed at the node in charge of processing the space-time planar graph, for example, the node i in Figure 23(a), after the graph is generated.

4.5. QoS Path switching

Among the available multiple paths, we use *only one path at a time* for each source-destination connection. Knowing each path's life periods, the source node can dynamically switch among these paths to send data packets. The goal of path switching in the proposed RMRV is to maintain data packet transmission with shorter delay so as to achieve QoS routing. The flowchart of the proposed QoS path switching is shown as Figure 24. The

guidelines of the QoS path switching are: (1) when the *current path* is about to disconnect, the source node switches to use another currently connected path, so as to reduce packet loss ; (2) the path with smallest estimated hop count is chosen first, which contributes to low packet delay [45]. For (1), when a path may become disconnected or connected can be known from its life periods, as introduced in Sections III-C and D. For (2), the hop count of each path is estimated as follows. We use the number of slots in RS_{ij}, h_{ij}, as the estimated hop count of RS_{ij}, where i represents path i and j = 1 to the number of RSs in path i. In RMRV, we let $h_{ij} = 2 + [(L_{ij} - 2*s) / (T_r/2)]$. L_{ij} is the length of RS_{ij} and s is a given constant, which is the length of the slot adjacent to neighbor RSs, for example, the length of slot X or W in Figure 23(a). $T_r/2$ is the slot length for the rest of slots, for example, the length of slot Y or Z in Figure 23(a). Then, the estimated hop count of a path i is derived as h_i, $h_i = \sum h_{ij}$. If multiple paths have the same estimated hop count, the path whose corresponding RREP packet was received first is chosen, because such a path has shorter delay. Then, the source node can use this chosen path to send data packets, until the path becomes disconnected.

For example, in Figure 25, the source node has three paths available to use. Path 1 is the first path from which the source node receives its RREP packet, so the source node chooses path 1 to send data first. However, path 1 is predicted to be disconnected at t_{11}, so the source node will switch to use path 2. Path 3 is not considered because it is also disconnected at time t_{11}. Then, path 2 is predicted be disconnected at t_{23}, and paths 1 and 3 are available to use. Because path 1 has a smaller estimated hop count than path 3, i.e. $h_1 < h_3$, which suggests it has shorter packet delay; thus the source node will choose path 1 to use.

Finally, to refresh the path life periods information, upon a periodic time of t_c timeout, the destination node would send a route update (RU) packet to the source node. The RU packet acts just like an RREP packet. It intersects the life periods of the RSs along the path. Besides, when the destination node moves to a new RS, an RU packet will also be generated

to report the new RS's life periods to the source node. Lastly, if route failure occurs, a route error (RERR) packet would be sent back to the source node, and the source node then deletes the corresponding path from its route table and uses another available path.

By our method, stable packet transmission can be achieved. The packet transmission is also QoS-aware which aims to decrease packet loss and packet delay. The proposed RMRV predicts paths' life periods, and thus can switch among the paths to avoid packet loss in advance. In addition, by our method we efficiently decrease route errors and the control overhead due to route errors and route re-discoveries is also significantly decreased. The reduction of route re-discoveries can also decrease packet collisions and thus benefit to QoS.

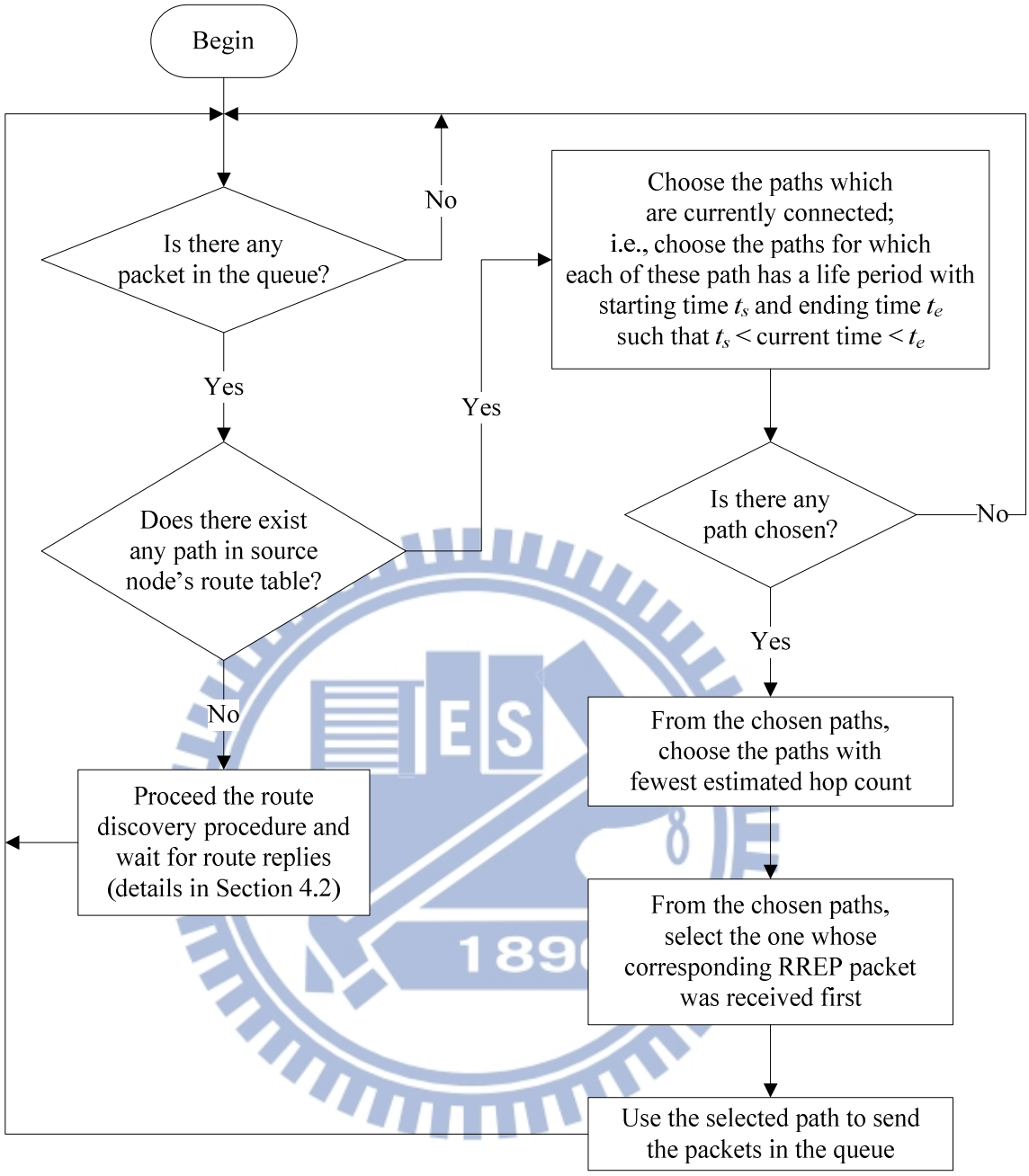


Figure 24. The flowchart of the proposed QoS path switching for a source node.

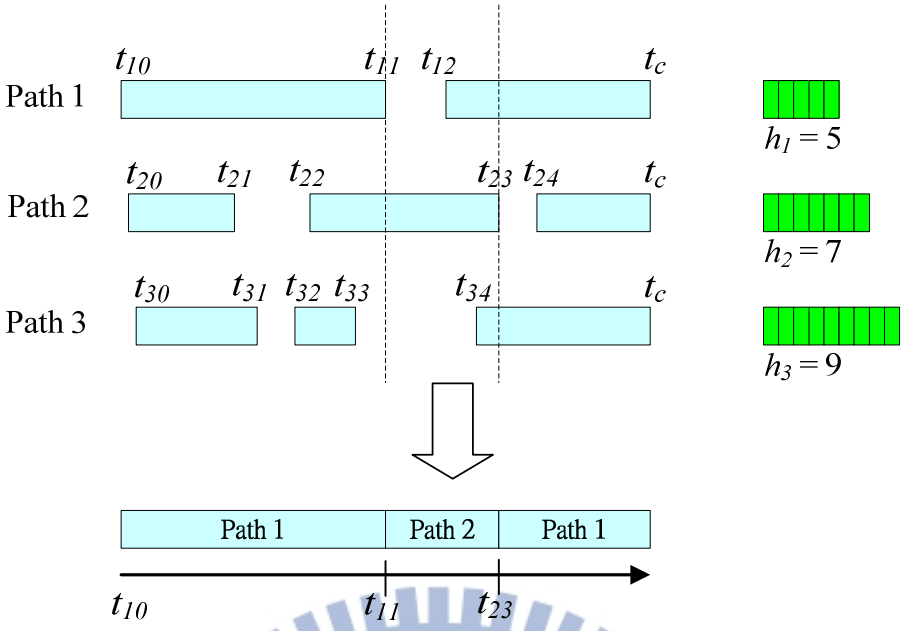


Figure 25. Path switching for better QoS. In RMRV, only one path is used for data transmission at a time. At first, path 1 is used and path 1 is predicted to be disconnected at t_{11}, and path 3 is also to be disconnected at t_{11}, so the source node will switch to use path 2. Then, path 2 is predicted be disconnected at t_{23}, and paths 1 and 3 are available to use. Because path 1 has a smaller estimated hop count than path 3, i.e. $h_1 < h_3$, the source node will choose path 1 to use.

4.6. Simulation results

We compare the proposed RMRV with RBVT-R, which is a representative road-based single path routing protocol. The figures of merit include packet delivery ratio, average end-to-end delay and control overhead, which will be defined later. We used QualNet 5.0 [20] for simulation. According to the IEEE 802.11p, the transmission is in the 5.9 GHz band. The transmitting power is set to 21.5 *dBm*, and the receiver sensitivity is set to -81.5 *dBm*. And the corresponding effective wireless range is 257 *m*. The MAC is CSMA/CA with RTS/CTS. And the MAC data rate is set to 12 Mbps. We used a total of 150 and 250 nodes distributed in a region of 1000 *m* x 1000 *m* in San Francisco's city map . The two-ray propagation model is

used in LOS (line of sight) areas. For NLOS areas, only nodes in adjacent roads are allowed for radio communication, where the distances of the nodes to the road intersection need to be within a distance of d_r. d_r is set to 80 *m*. The node speed is in the range of [5 *m*/*s*, 20 *m*/*s*]. Node mobility traces were generated using VanetMobiSim [10]. VanetMobiSim provides lane changing, car-following, intersection management and traffic lights models, etc. The simulation time is 300 *s*. Packets are generated in a constant bit rate (CBR) at the source node. The default packet sending rate is 4 *packets/s*. The packet size is 512 bytes. We used 2 ~ 20 CBR connections concurrently in the simulation. For each CBR connection, m_u is the maximum number of paths in the route table at the source node. m_u is set to 2 ~ 5. Note that among the multiple paths, we use *only one path at a time* for the current CBR connection. o_u is the maximum number of overlapping RSs allowed among the multiple paths without considering the beginning and ending RSs. o_u is set to 3. Periodic route update timeout t_c is set to 5 *s*. Note that RBVT-R has modified IEEE 802.11 RTS/CTS frames in the MAC layer, but in this dissertation we focus on performance enhancements of multipath routing to single-path routing. Therefore, for fair comparison, we only implemented the route discovery and route maintenance algorithms of RBVT-R. As to data packet transmission, for both RMRV and RVBT-R, if a route error occurs and there is no available connected path, we buffer the data packets at the source node until new paths available by path re-discovery.

4.6.1. Packet delivery ratio comparison

The comparison of *packet delivery ratio* for the two routing algorithms is shown in Figure 26. The packet delivery ratio is defined as the ratio of the number of data packets received at the destination to the number of data packets sent from the source. The packet delivery ratio of the proposed RMRV outperforms that of RBVT-R by 16.9%, on average. This is because that RMRV periodically predicts the multiple paths' future life periods and

can switch to use another path before the current path disconnects; therefore the packet loss is reduced. In addition, RMRV incurs route re-discovery less frequently so that the extra generated control traffic is much reduced. Thus, packet collisions are also reduced and the packet delivery ratio is then enhanced.

The packet delivery ratio of using different number of multiple paths for RMRV is also shown in Figure 26. The increase of available paths benefits the increase of packet delivery ratio. This is because that if there are more available paths maintained in the route table, the data transmission will not be disrupted with high probability. Also, route re-discovery would be invoked less frequent; thus the control traffic due to route request packets incurred by route re-discovery will be greatly reduced. Less control traffic bursts can also reduce data connections suffering from packet collisions and possible packet loss. However, when the number of nodes increases to 250 nodes, as shown in Figure 26(b), the improvement of using more paths in RMRV is not as significant as that in Figure 26(a). This is because that with high node density, RS connectivity is also high, and thus a path would not become disconnected easily. As a result, giving additional available paths brings no significant improvement.

4.6.2. End-to-end delay comparison

The comparison of *average end-to-end delay* for the two routing algorithms is shown in Figure 27. The end-to-end delay is defined as the average delay of a packet, from the packet was generated at the source node until it reaches the destination node, including the time the packet was queued at the source node. The average end-to-end delay of RMRV is 35.3% lower than that of RBVT-R, on average, as shown in Figure 27. This is because that when the current path is about to be disconnected, RMRV can switch to use another connected path immediately, but RBVT-R has to wait route re-discovery to continue data packet transmission.

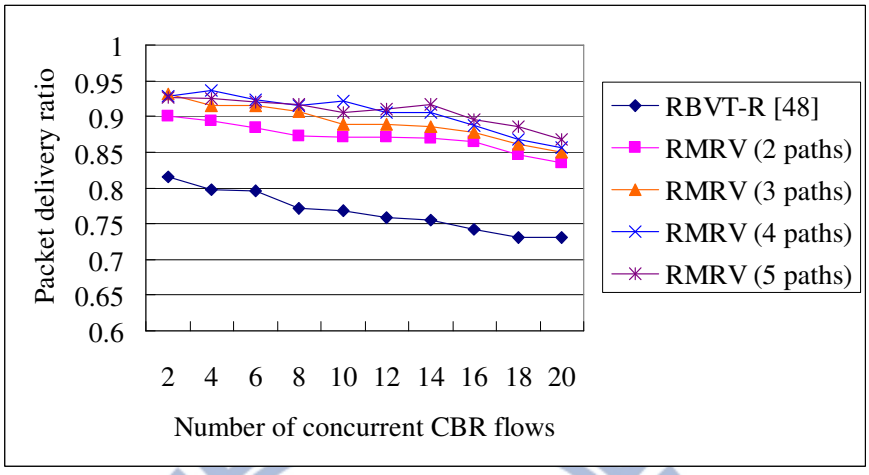
The end-to-end delay of using different number of paths for RMRV is also shown in Figure 27. With more paths available to choose to switch to, there is less need to pause data packet transmission due to route re-discovery when there is no connected path to use, and thus resulting in lower end-to-end delay. In addition, when the number of nodes increases to 250 nodes, as shown in Figure 27(b), the end-to-end delay significantly decreases. This is because that with high node density, paths would not disconnect easily. Therefore, the need for route re-discovery is much reduced; as a result, the end-to-end delay is also much reduced.

4.6.3. Control overhead comparison

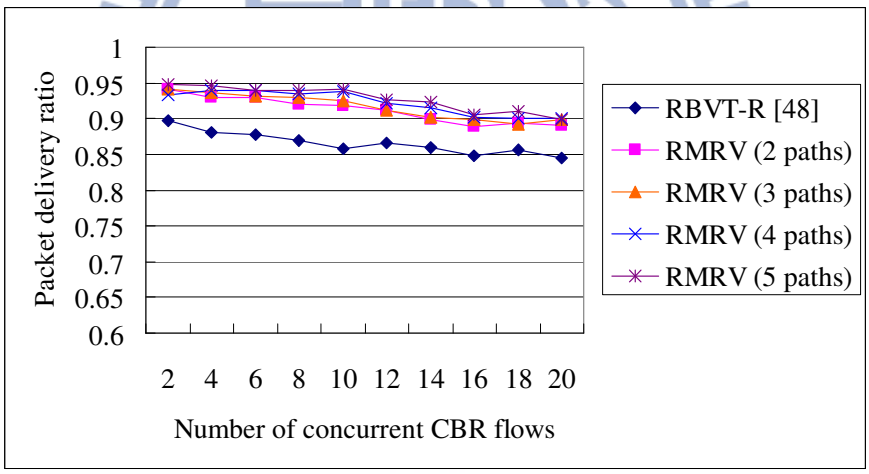
The comparison of control overhead for the two routing algorithms is shown in Figure 28. Control overhead is defined as the ratio of control bytes generated by all nodes in the network to the data bytes received by the destination. As shown in Figure 28, the control overhead of RMRV is 45.8% lower than that of RBVT-R, on average. In the initial route discovery, both RMRV and RBVT-R generate a similar number of RREQ packets because they both flood the RREQ packet to the whole network. The difference between them is that RMRV has to send multiple RREP packets of the discovered multiple paths from destination to source, and RBVT-R sends only one RREP packet. However, the RREP packets are propagated by unicast; thus the incurred traffic is very small. Remind that when a path failure occurs, RBVT-R has to proceeds route discovery immediately. Route discovery floods RREQ packets to the whole network, producing large control traffic. In contrast, RMRV can use another available path instead. Therefore, the route re-discovery in RMRV is less frequent and it results in less control overhead.

The control overhead of using different number of multiple paths for RMRV is also shown in Figure 28. With more paths available to switch to, route re-discovery is less frequent and thus it results in less control overhead. When the number of nodes increases to 250 nodes,

as shown in Figure 28(b), control overhead is further decreases. This is because the need for route re-discovery is much reduced under high node density due to high network connectivity.

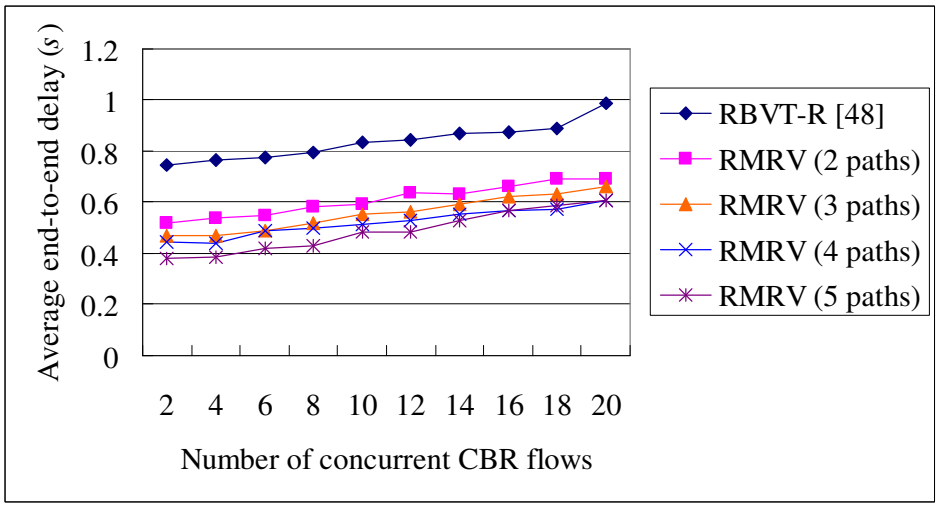


(a) 150 nodes case

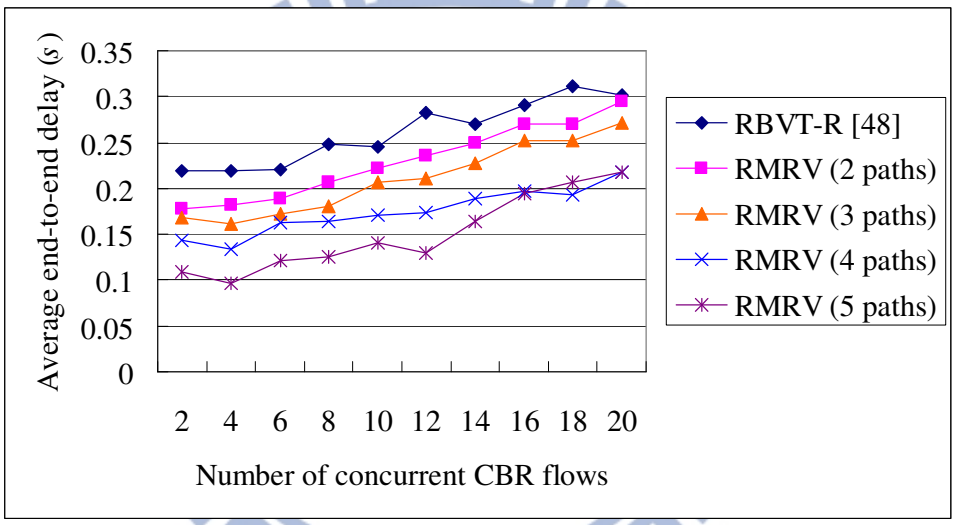


(b) 250 nodes case

Figure 26. Comparison of packet delivery ratio. (a) 150 nodes case, and (b) 250 nodes case.

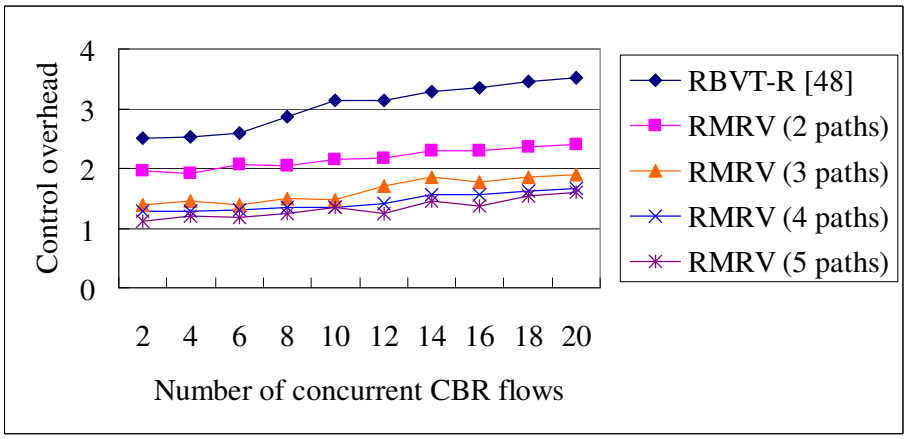


(a) 150 nodes case

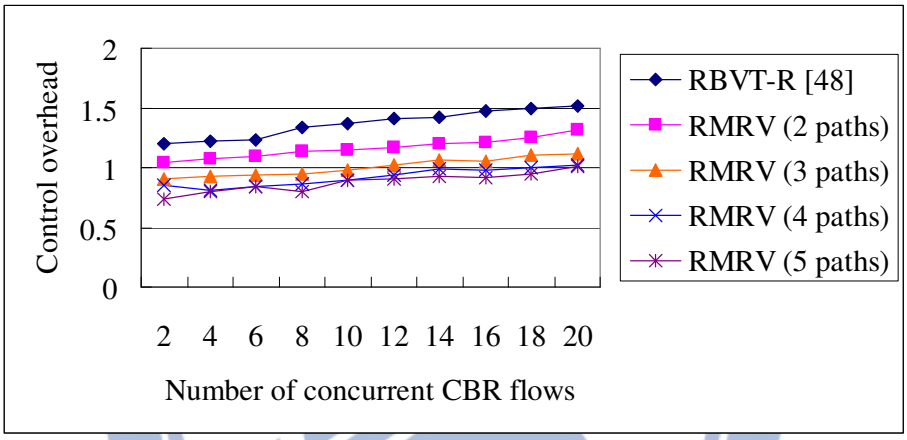


(b) 250 nodes case

Figure 27. Comparison of average end-to-end delay. (a) 150 nodes case, and (b) 250 nodes case.



(a) 150 nodes case



(b) 250 nodes case

Figure 28. Comparison of control overhead. (a) 150 nodes case, and (b) 250 nodes case.

Chapter 5

Conclusion and future work

We have explored the feasibility of live multimedia streaming by overlay multicast in urban VANETs. To adapt to high mobility and full of obstacles in urban VANETs, we have presented an effective dynamic overlay multicast in VANETs (OMV). The proposed OMV is QoS-satisfied considering both packet loss rate and end-to-end delay. We have also simulated obstacles in urban VANETs so as to reflect real world scenarios. We have evaluated the proposed OMV in urban VANETs with obstacles using two real video clips to demonstrate the feasibility of the OMV for real videos. Evaluation results show that comparing the proposed OMV to Qadri *et al.*'s work, the packet loss rate is reduced by 27.1% and the end-to-end delay is decreased by 11.7%, with a small control overhead of 2.1%, on average. Comparing the proposed OMV for tree overlays to ALMA, the packet loss rate is reduced by 7.1% and the end-to-end delay is decreased by 13.1%. Due to high impact of obstacles on performance in urban VANETs, we have also investigated feasible stream rates under different overlay sizes and road section sizes. The future work includes extending a multicast overlay that allows multiple overlay nodes concurrently being source nodes and receiving nodes (e.g., for live video conferencing), enhancing the performance of a multicast overlay that integrates with the underlying routing protocol, and investigating the feasibility of allowing more than two parents for a child to have high QoS.

We have also presented a road-based QoS-aware multipath routing protocol for urban VANETs (RMRV). The proposed RMRV can find multiple paths and estimate the paths'

future life periods for QoS path switching. A novel space-time planar graph approach has been proposed to predict the connectivity of each road section in a path, and thus a path's future life periods can be derived. Simulation results have shown that the proposed RMRV has significant improvement on the packet delivery ratio, average end-to-end delay and control overhead, over a representative road-based single path routing protocol, RBVT-R. To the best of our knowledge, there is no existing road-based multipath routing protocol for VANETs. The proposed RMRV is very suited to urban VANETs with high mobility and dense obstacles. The future work includes (1) porting the RS life periods prediction mechanism to the cloud so as to avoid redundant RSs life periods calculation due to duplicate RSs among different pairs of source and destination, (2) in the space-time planar graph, instead of assuming that every vehicle keeps moving in a constant speed, considering the vehicle's behaviors of acceleration and braking, (3) dynamically adjusting the interval of route updates according to node density, (4) cross-layer design with additional information from lower layers such as the bit error rate, and (5) using multiple paths concurrently to speed up data transmission.

References

[1] N. N. Qadri, M. Fleury, M. Altaf and M. Ghanbari, "Multi-source video streaming in a wireless vehicular ad hoc network," *IET Communications*, vol. 4, issue 11, pp. 1300-1311, 2010.

[2] U. Lee, J.-S. Park, J. Yeh, G. Pau and M. Gerla, "CodeTorrent: a content distribution using network coding in VANET," in *Proc. of the First ACM Workshop on Decentralized Resource Sharing in Mobile Computing and Networking*, pp. 1 - 5, 2006.

[3] J. -S Park, U. Lee, S. Y. Oh and M. Gerla, "Emergency related video streaming in VANET using network coding," in *Proc. of the Third International Workshop on Vehicular Ad Hoc Networks, ACM,* pp. 102 - 103, 2006.

[4] M. Abuelela and S. Olariu, "ZIPPER: a zero-Infrastructure peer-to-peer system for VANET," in *Proc. of the Third ACM Workshop on Wireless Multimedia Networking and Performance Modeling*, pp. 2-8, 2007.

[5] Y.-C. Chu and N.-F. Huang, "Delivering of live video streaming for vehicular communication using peer-to-peer approach," in *Proc. of the MOVE Workshop, INFOCOM,* May 2007.

[6] F. Soldo, C. Casetti, C.-F. Chiasserini and P. Chaparro, "Streaming media distribution in VANETs," in *Proc. of the IEEE GLOBECOM*, pp. 1-6, 2008.

[7] Q. Li, Y. Andreopoulos, and M. van der Schaar, "Streaming-viability analysis and packet scheduling for video over in-vehicle wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3533-3549, 2007.

[8] M. Guo, M. H. Ammar and E. W. Zegura, "V3: a vehicle-to-vehicle live video streaming architecture," in *Proc. of the Third IEEE International Conference on Pervasive Computing and Communications*, pp. 171 - 180, 2005.

[9] Y. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, pp. 307 - 321, 2000.

[10] M. Fiore, J. Harri, F. Filali and C. Bonnet, "Vehicular mobility simulation for VANETs," in *Proc. of the 40th Annual Simulation Symposium*, pp. 301 - 309, 2007.

[11] N.N. Qadri, M. Altaf, M. Fleury, M. Ghanbari, and H. Sammek, "Robust video streaming over an urban VANET," in *Proc. of the Second IEEE Int. Workshop on Selected Topics in*

Mobile and Wireless Computing, Marrakech, pp. 429 - 434, 2009.

[12] J. Nzouonta, .N. Rajgure, G. Wang, and C. Borcea, "VANET routing on city roads using real-time vehicular traffic information," *IEEE Transactions on Vehicular Technology*, vol. 58 , no. 7, pp. 3609 - 3626, 2009.

[13] Y.H. Ho, A.H. Ho, and K.A. Hua, "Routing protocols for inter-vehicular networks: a comparative study in high-mobility and large obstacles environments," *Computer Communications*, vol. 31, pp. 2767 - 2780, 2008.

[14] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. of the 25th IEEE INFOCOM*, pp. 1-12, 2006.

[15] T. Rasheed, M. Jerbi, S.M. Senouci, and Y. Ghamri-Doudane, "Towards efficient geographic routing in urban vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5048 - 5059, November 2009.

[16] C. Gui and P. Mohapatra, "Efficient overlay multicast for mobile ad hoc networks," in *Proc. of the IEEE WCNC*, pp.1118 - 1123, 2003.

[17] M. Ge, S.V. Krishnamurthy and M. Faloutsos, "Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol," *Ad Hoc Networks*, vol. 4, issue 2, pp. 283 - 300, 2006.

[18] C. Gui and P. Mohapatra, "Efficient overlay multicast for mobile ad hoc networks," in *Proc. of the IEEE WCNC*, 2003, pp. 1118 - 1123.

[19] K. Kim and S.-H. Kim, "A novel overlay multicast protocol in mobile ad hoc networks: design and evaluation," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 6, Nov. 2005.

[20] Qualnet, http://www.scalable-networks.com/.

[21] D. Jiang and L. Delgrossi, "IEEE 802.11p: towards an international standard for wireless access in vehicular environments," in *Proc. of the IEEE Vehicular Technology Conference*, pp. 2036 - 2040, 2008.

[22] Z. Yang, M. Li and W. Lou, "CodePlay: live multimedia streaming in VANETs using symbol-level network coding," in *Proc. of the 18th IEEE International Conference on Network Protocols*, pp. 223 - 232, 2010.

[23] S.-Y. Wang, P.-F. Wang, Y.-W. Li and L.-C. Lau, "Design and implementation of a more realistic radio propagation model for wireless vehicular networks over the NCTUns network simulator," *in Proc. of the IEEE Wireless Communications and Networking Conference* (*WCNC*), pp. 1937 – 1942, 2011.

[24] N.N. Qadri, M. Fleury, M. Altaf and M. Ghanbari, "P2P layered video streaming over

wireless ad hoc networks," in *Proc. of the 5th International ICST Mobile Multimedia Communications Conference*, 2009.

[25] K. Chen and K. Nahrstedt, "Effective location-guided tree construction algorithms for small group multicast in MANET," in *Proc. of the IEEE INFOCOM*, 2002, pp. 1180 - 1189.

[26] ITU, "Subjective video quality assessment methods for multimedia applications," *ITU-T Recommendations*, p. 910, 1996.

[27] C.-O. Chow and H. Ishii, "Enhancing real-time video streaming over mobile ad hoc networks using multipoint-to-point communication," *Computer Communication*, vol. 30, pp. 1754–1764, 2007.

[28] J. He, A. Chaintreau, C. Diot, "A performance evaluation of scalable live video streaming with nano data centers," *Computer Networks*, vol. 53, issue 2, pp. 153 - 167, 2009.

[29] P. Symes, "Digital Video Compression," McGraw-Hill, USA, 2001.

[30] F. Wang, J. Liu, Y. Xiong, "On node stability and organization in peer-to-peer video streaming systems," *IEEE Systems Journal*, vol. 5, issue 4, pp. 440 - 450, 2011.

[31] F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate and P. Manzoni, "Realistic radio propagation models (RPMs) for VANET simulation," in *Proc. of the IEEE Wireless Communications and Networking Conference*, 2009, pp. 1155- 1160, 2009.

[32] T. S. Rappaport, "Wireless communications, principles and practice," Prentice Hall, 1996.

[33] A. Panayides, M. S. Pattichis, Constantinos S. Pattichis, C. P. Loizou, M. Pantziaris, and A. Pitsillides, "Atherosclerotic plaque ultrasound video encoding, wireless transmission, and quality assessment using H.264," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, 2011.

[34] T. T. Thai, J. Lacan and H. Meric, "Error tolerance schemes for H.264/AVC: an evaluation," in *Proc. of IEEE Consumer Communications & Networking Conference* (*CCNC*), pp. 571 – 575, 2012.

[35] K. Tan, "A new error resilience scheme based on FMO and error concealment in H.264/AVC," in *Proc. of 2011 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 1057-1060, 2011.

[36] F. Zhu, W. Zhang, N. Yu, J. Xu and G. Wu, "Adaptive error resilient coding based on FMO in wireless video transmission," in *Proc. of Third International Conference on Multimedia Information Networking and Security*, pp. 609 – 612, 2011.

[37] FFmpeg, http://ffmpeg.org.

[38] MP4Box, http://gpac.wp.mines-telecom.fr/mp4box/.

[39] YUV video sequences, http://trace.eas.asu.edu/yuv/index.html.

[40] Jens-Rainer Ohm. Bildsignalverarbeitung fuer multimedia-systeme. Skript, 1999.

[41] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid - a framework for video transmission and quality evaluation," in *Proc. of 13th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, USA, 2003.

[42] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, pp. 969-988, 2006.

[43] C.-S. Wu, S.-C. Hu and C.-S. Hsu, "Design of fast restoration multipath routing in VANETs", in *Proc. of International Computer Symposium* (ICS), pp. 73-78, 2011.

[44] T. Dreibholz, M. Becke, E. P. Rathgeb and M. Tuxen, "On the use of concurrent multipath transfer over asymmetric paths," in *Proc. of the IEEE Global Communications Conference* (GLOBECOM), pp. 1-6, 2010.

[45] H. Saleet, R. Langar, K. Naik, R. Boutaba, A. Nayak and N. Goel, "Intersection-based geographical routing protocol for VANETs: a proposal and analysis," *IEEE Transactions on Vehicular Technology*, vol. 60, issue 9, pp. 4560-4574, Nov. 2011.

[46] M. Jerbi, S.-M. Senouci, R. Meraihi and Y. Ghamri-Doudane, "An improved vehicular ad hoc routing protocol for city environments," in *Proc. of IEEE International Conference on Communications* (ICC), pp. 3972-3979, 2007.

[47] K. Lee, M. Le, J. Haerri and M. Gerla, "LOUVRE: Landmark overlays for urban vehicular routing environments," in *Proceedings of the 68th IEEE Vehicular Technology Conference* (VTC), pp. 1-5, 2008.

[48] J. Nzouonta, N. Rajgure, G. Wang and C. Borcea, "VANET routing on city roads using real-time vehicular traffic information," *IEEE Transactions on Vehicular Technology*, vol. 58, issue 7, pp. 3609-3626, 2009.

[49] H. Rongxi ,H. Rutagemwa and S. Xuemin, "Differentiated reliable routing in hybrid vehicular ad-hoc networks," in *Proc. of Int. Conference on Communications* (ICC), pp. 2353-2358, May 2008.

[50] M. Fiore, J. Harri, F. Filali and C. Bonnet, "Vehicular mobility simulation for VANETs," in *Proc. of the 40th Annual Simulation Symposium*, pp. 301-309, 2007.

[51] S. Bitam and A. Mellouk, "QoS swarm bee routing protocol for vehicular ad hoc networks," in *Proc. of Int. Conference on Communications* (ICC), pp. 1-5, June 2011.

[52] Y. Gongjun, D.B. Rawat and B.B. Bista, "Provisioning vehicular ad hoc networks with

quality of service," in *Proc. of International Conference on Broadband, Wireless Computing, Communication and Applications* (BWCCA), pp. 102-107, 2010.

[53] Z. Mo, H. Zhu, K. Makki and N. Pissinou, "MURU: A multi-hop routing protocol for urban vehicular ad hoc networks," in *Proc. of the 3rd Annual Int. Conference on Mobile and Ubiquitous Systems*, pp. 1-8, 2006.

[54] 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* (ACM MobiCom '00), pp. 243-254, 2000.

[55] C. Lochert, M. Mauve, H. Füßler and H. Hartenstein, "Geographic Routing in City Scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, issue 11, pp. 69 - 72, 2005.

[56] U.S. Department of Transportation, National Highway Traffic Safety Administration, "Vehicle Safety Communications Project V - Final Report", *Report. DOT HS 810 591*, Apr. 2006.

[57] J.B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, issue 7, pp. 1162 - 1182, 2011.

[58] M. Abuelela and S. Olariu, "Traffic-adaptive packet relaying in VANET," in *Proc. of ACM Workshop on Vehicular Ad Hoc Networks*, pp. 77 - 78, 2007.

[59] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *http://www.ietf.org/rfc/rfc3561.txt*, July 2003. RFC 3561.

Publication list

Journal papers

1. **Yi-Ling Hsieh**, Kuochen Wang, "Dynamic Overlay Multicast for Live Multimedia Streaming in Urban VANETs," ***Computer Networks***, vol. 56, issue 16, Nov. 2012, pp. 3609-3628, Place: USA. **(SCI)**
2. Shen-Hai Shee, Tzu-Chien Chang, Kuochen Wang, **Yi-Ling Hsieh**, "Efficient Color-theory-based Dynamic Localization for Mobile Wireless Sensor Networks," ***Wireless Personal Communications***, vol. 59, no. 2, Feb. 2010, pp. 375-396, Place: SpringerLink. **(SCI Expanded, EI)**
3. Tai-Jung Chang, Kuochen Wang, **Yi-Ling Hsieh**, "A Color-theory-based Energy Efficient Routing Algorithm for Mobile Wireless Sensor Networks," ***Computer Networks***, vol. 52, no. 3, Feb. 2008, pp. 531-541, Place: USA. **(SCI)**

Conference papers

1. **Yi-Ling Hsieh**, Kuochen Wang, "A Road-based QoS-aware Multipath Routing for Urban Vehicular Ad Hoc Networks," in *Proceedings of IEEE GLOBECOM* (**GLOBECOM 2012**), Dec. 2012, Place: USA.
2. Min-Hsuan We, Kuochen Wang, **Yi-Ling Hsieh**, "A Reliable Routing Scheme Based on Vehicle Moving Similarity for VANETs," in *Proceedings of the 2011 IEEE Region 10 Conference* (TENCON), Nov. 2011, Place: Indonesia.
3. Hsuan-Fu Ho, Kuochen Wang, **Yi-Ling Hsieh**, "Resilient Video Streaming for Urban VANETs," in *Proceedings of the 7th Workshop on Wireless Ad Hoc and Sensor Networks* (WASN), Sep. 2011, Place: Taiwan.
4. **Yi-Ling Hsieh**, Kuochen Wang, "Road Layout Adaptive Overlay Multicast for Urban Vehicular Ad Hoc Networks," in *Proceedings of the 73rd IEEE Vehicular Technology Conference* (**VTC**), May 2011, Place: Hungary.
5. Chia-Yi Liu, Kuochen Wang, **Yi-Ling Hsieh**, "Efficient Push-Pull Based P2P Multi-streaming Using Application Level Multicast," in *Proceedings of the 21th IEEE*

International Symposium on Personal, Indoor and Mobile Radio Communications (**PIMRC**), Sep. 2010, Place: Turkey.

6. Bo-Wei Li, Kuochen Wang, **Yi-Ling Hsieh**, "A Hierarchical Social Network-based P2P SIP System for Mobile Environments," in Proceedings of the *21th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (**PIMRC**), Sep. 2010, Place: Turkey.
7. Wei-Cheng He, Kuochen Wang, **Yi-Ling Hsieh**, "Dependable Peer to Peer Multi-Streaming Using DHT-based Application Level Multicast," in *Proceedings of the 9th IASTED International Conference on Wireless and Optical Communications* (WOC), July 2009, Place: Canada.
8. Tzu-Chien Chang, Kuochen Wang, **Yi-Ling Hsieh**, "Enhanced Color-theory-based Dynamic Localization in Mobile Wireless Sensor Networks," in *Proceedings of the 2007 IEEE Wireless Communications and Networking Conference* (**WCNC**), pp. 3064-3069, March 2007, Place: China.
9. Hsing-Yu Lu, Kuochen Wang, **Yi-Ling Hsieh**, "Power-Efficient MAC Protocol for VoIP Traffic over IEEE 802.11e WLAN," in *Proceedings of the International Computer Symposium: Workshop on Computer Networks*, Dec. 2006, Place: Taiwan.
10. Tai-Jung Chang, Kuochen Wang, **Yi-Ling Hsieh**, "A Color-theory-based Energy Efficient Routing Algorithm for Wireless Sensor Networks," in *Proceedings of Workshop on Wireless, Ad Hoc, Sensor Networks* (WASN), Aug. 2006, Place: Taiwan.
11. **Yi-Ling Hsieh**, Kuochen Wang, "Efficient Localization in Mobile Wireless Sensor Networks," in *Proceedings of the 2006 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing* (SUTC 2006), June 2006, Place: Taiwan.
12. Shen-Hai Shee, Kuochen Wang, **Yi-Ling Hsieh**, "Color-theory-based Dynamic Localization in Mobile Wireless Sensor Networks," in *Proceedings of Workshop on Wireless, Ad Hoc, Sensor Networks* (WASN), Aug. 2005, Place: Taiwan. (**Received the Best Paper Award**)