Computer Networks 56 (2012) 3609–3628

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Dynamic overlay multicast for live multimedia streaming in urban VANETs

Yi-Ling Hsieh, Kuochen Wang *

Department of Computer Science, National Chiao Tung University, Taiwan

ARTICLE INFO

Article history:
Received 18 January 2012
Received in revised form 15 June 2012
Accepted 24 July 2012
Available online 31 July 2012

Keywords:
Dynamic overlay multicast
Multimedia streaming
QoS
Urban VANETs

ABSTRACT

Infotainment service has been a foreseeing trend in VANETs (Vehicular Ad Hoc Networks), and multimedia streaming has a high potential in VANET infotainment service. This paper considers 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.

1. Introduction

The feasibility of VANETs (Vehicular Ad Hoc Networks) is promising due to establishment of ITS (Intelligent Transport System) infrastructure, availability of OBUs (on-board units) for vehicles, and emerging communication standards including IEEE 802.11p and IEEE 1609.1-4 for inter vehicular communications. Through inter-vehicle communications, vehicles can exchange data with each other, including multimedia data. Thus, infotainment service in VANETs is a foreseeing trend. More and more studies reveal its feasibility and applicability [2–8,22]. The delivered

* Corresponding author. Tel.: +886 3 5131363.
E-mail addresses: ylhsieh@cs.nctu.edu.tw (Y.-L. Hsieh), kwang@cs.nctu.edu.tw (K. Wang).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.07.011

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. Live multimedia streaming with overlay multicast in VANETs

This paper 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. This paper focuses 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 Fig. 1, 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 [12–15]. An overlay itself just needs to focus on application layer optimization.

1.2. Dynamic overlay in urban VANETs

We consider how to adapt an overlay to urban VANETs for live streaming. 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 [16–19,25], 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.

1.3. Contribution of this paper

We propose a dynamic overlay multicast design suited for multimedia streaming in urban VANETs. To the best

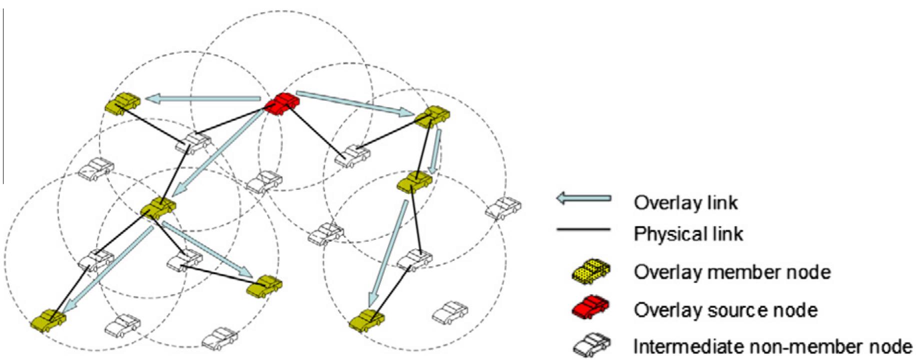


Fig. 1. An example overlay multicast in a VANET.

of our knowledge, dynamic overlay multicast for multimedia streaming has not been studied in VANETs literature.

2. Related work

Existing literatures on multimedia streaming approaches for VANETs are classified in Fig. 2. We review these literatures in the following, and discuss their feasibility of multimedia streaming to a group of nodes in urban VANETs.

2.1. 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.

2.2. 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.

2.3. 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 re-

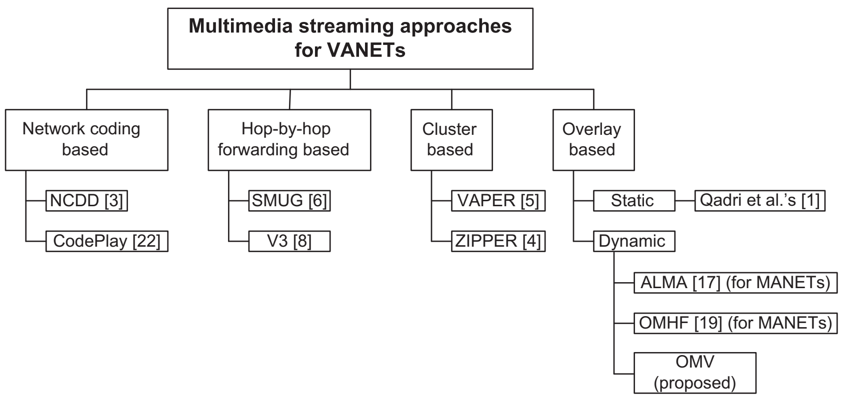


Fig. 2. Classification of existing multimedia streaming approaches for VANETs.

trieve 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.

2.4. 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 [16–19,25], 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.

3. Proposed overlay multicast in urban VANETs

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

This paper 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 Fig. 3a and 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.

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	Application layer static mesh overlay	Application layer dynamic tree overlay (for MANETs)	Application layer dynamic mesh overlay
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 considered	No	No	No	No	No	No	Yes
Obstacles evaluated	No	No	No	No	Yes	No	Yes

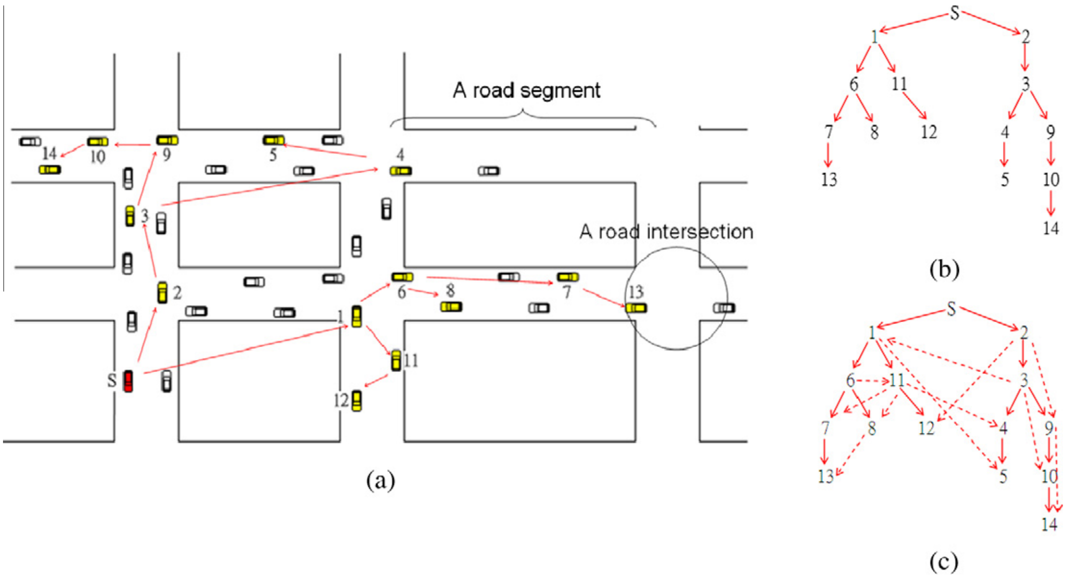


Fig. 3. An illustration of a multicast overlay in urban VANETs: (a) an overlay formed by interested vehicles, in urban roads; (b) the corresponding overlay tree; and (c) expanding the tree to a mesh so that nodes are allowed to have two parents.

As vehicles move rapidly, an overlay tree need to change its structure to maintain QoS. For example, as shown in Fig. 3a, 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 Fig. 3c. 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.

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 Fig. 4. 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.

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

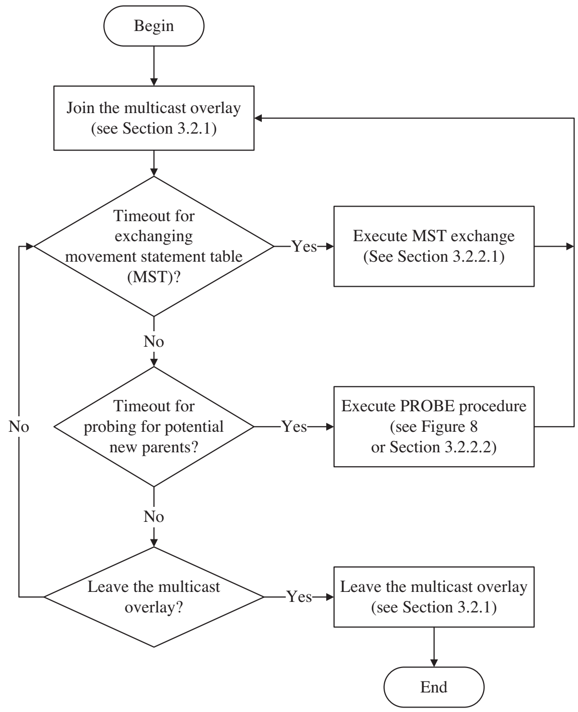


Fig. 4. Flowchart of the life cycle of a group node in the proposed multicast overlay.

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 5–10 s 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. Fig. 5 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 Fig. 5b. 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

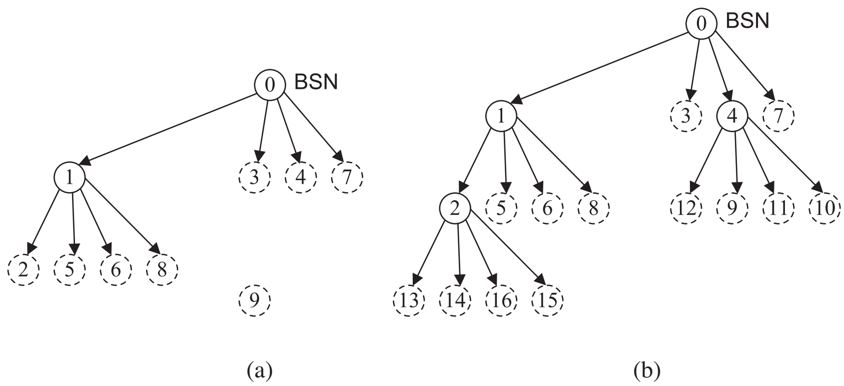


Fig. 5. Imbalanced tree problem due to JOIN requests bottleneck in the BSN. (a) The BSN receives nodes 2–9's JOIN requests, but currently only nodes 0 and 1 can be candidate parents. (b) Nodes 2 and 4 finish the join procedure. Then the BSN replies nodes 9–15 of their candidate parents as nodes 2 or 4.

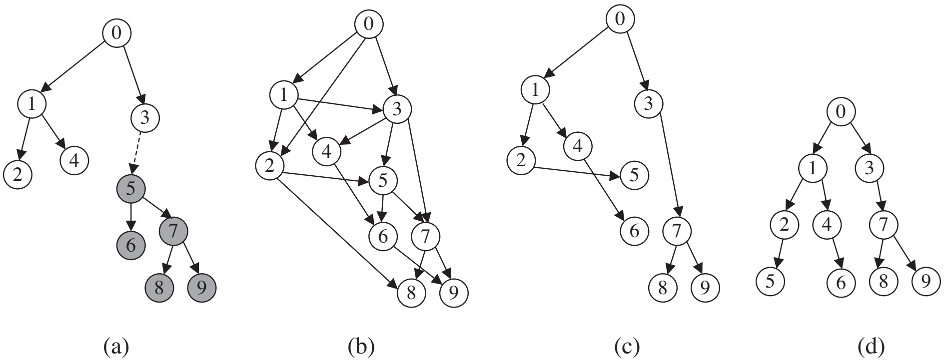


Fig. 6. Solution to the imbalanced tree problem, as shown in Fig. 5a. (a) All requesting nodes are allowed to be assigned by the BSN to be candidate parents. Note that join failure of a whole subtree may occur, such as node 5 and its descendents. (b) In addition to (a), let the BSN provide two candidate 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.

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 Fig. 6a. 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 6–9) 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 Fig. 6b, 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. Fig. 6c shows the final tree, and is redrawn as shown in Fig. 6d. 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 Section III-B2b.

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 Fig. 7. Assume node A

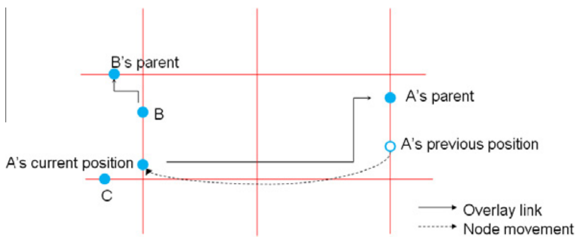


Fig. 7. Choosing a better parent.

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 3.2.2.1, and then introduce the PROBE procedure in Section 3.2.2.2.

3.2.2.1. 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.

3.2.2.2. Probing for a new parent. As shown in Fig. 7, 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 × (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 × (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.

3.2.2.2.1. Proposed PROBE procedure. As shown in Fig. 8, 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 m) from node A. These nodes form a set P, such that size $(P) \leqslant$ 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, $R' = \{x_i | d_i + t_i < T_B, \forall x_i \in R\}$. 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, node A sends a RESIGN message to node A's current parent. If node z rejects, node A removes node z from R' and chooses another node with 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.

Fig. 9 uses an example to illustrate the PROBE procedure. Node A looks up its MST and finds that 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 2–4 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 2–4, 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.

3.2.2.2.2. 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.

3.2.2.2.3. 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 *min heap*. That is, the packet loss rate of a node is lower than that of its descendants, in a min 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 min 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 descendents 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 paper, 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,

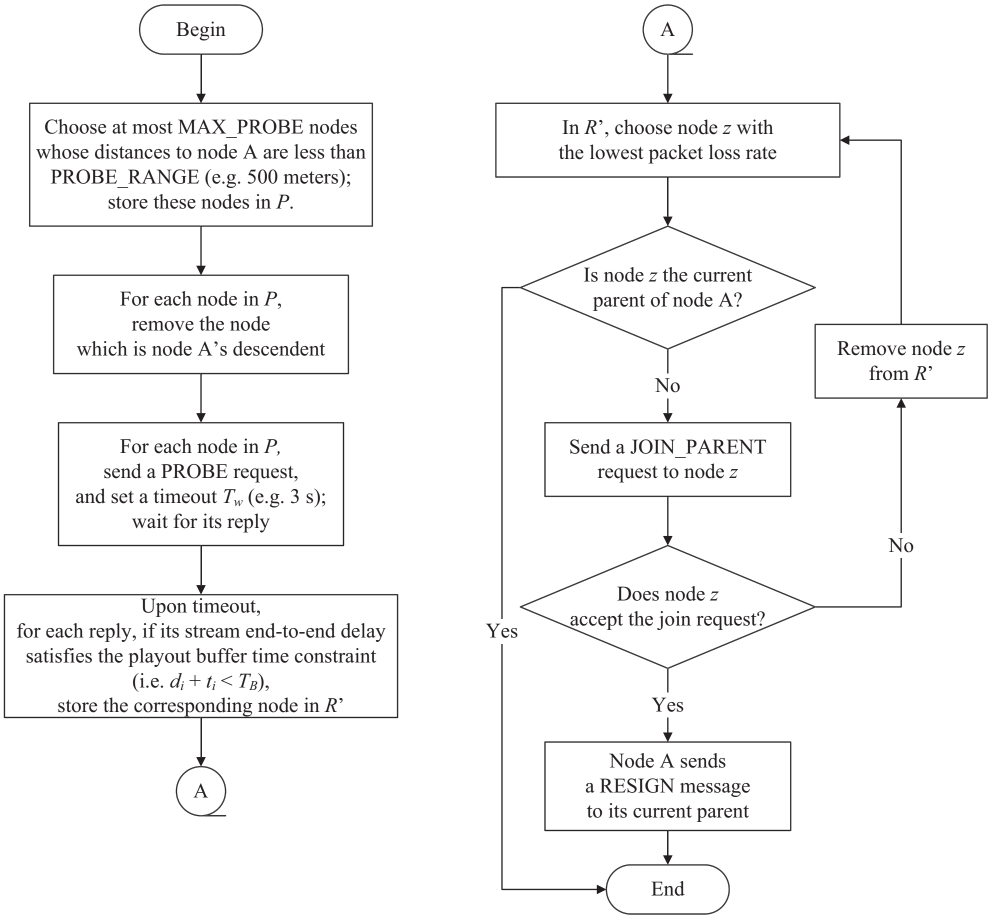


Fig. 8. The PROBE procedure for node A to find and switch to a better new parent.

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.

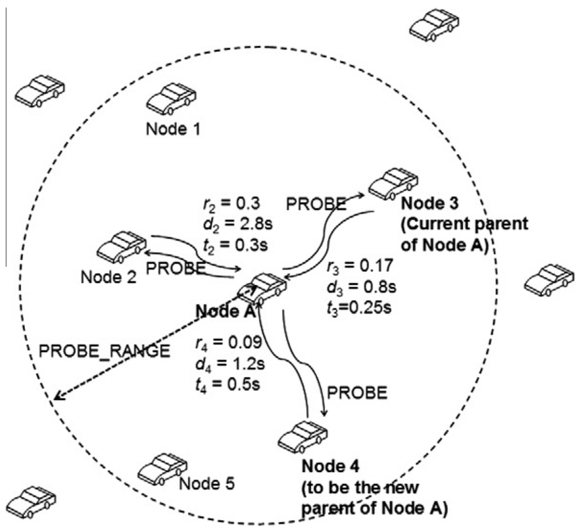


Fig. 9. An example to illustrate the PROBE procedure.

3.2.4. Handling 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 Fig. 8. The PROBE procedure is periodically invoked. Therefore, for nodes disconnected from their parents, or for the children that their parent unexpectedly leaves, 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.

4. 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 method 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.

4.1. Simulation settings

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 $\times$ 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 $\times$ 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 paper we assume the radio is totally blocked by obstacles [13]. In line-of-sight areas, the two-ray ground propagation model [20,32] 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. The settings of city maps and vehicle mobility are shown in Table 2. The default map size is 1000 m $\times$ 1000 m. We have three map scenarios, maps 1–3, as shown in Table 2. The number of traffic lights is in proportion to the number of road intersections in each map.

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 3. The playout buffer time for live streaming is set as 3 s [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.

4.2. Simulation results

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.

Table 2
Settings of city maps and vehicle mobility.

Settings of city maps [10]	
Field size	Default: 1000 m × 1000 m
Map layout	*UserGraph* [10] as a grid map with uniform road section size
Road section size	100 m
Default map	1000 m × 1000 m with road section size of 100 m
Map scenario 1 (map 1)	1200 m × 1200 m with road section size of 100 m
Map scenario 2 (map 2)	1200 m × 1200 m with road section size of 200 m
Map scenario 3 (map 3)	1200 m × 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

Table 3
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 verlay 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

Table 4
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) *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.

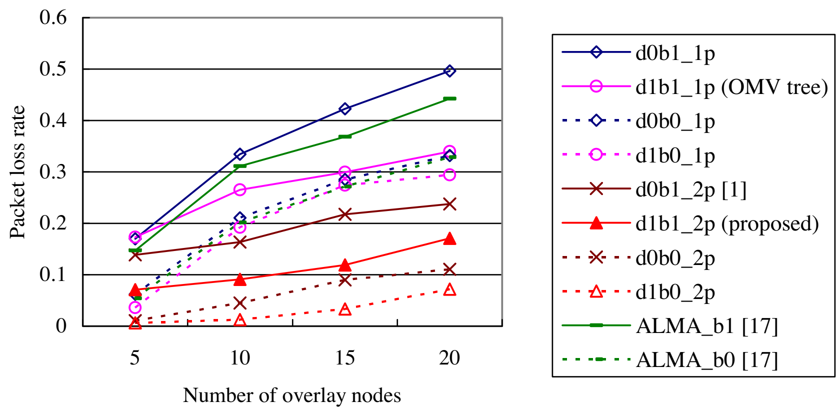
We evaluate the above three metrics for different approaches, under overlay sizes from 5 to 20 nodes. In the simulation results, the following labels are used to represent different cases of overlays, as shown in Table 4. For example, d0b1_1p represents an obstacle-prone static tree overlay. Note that the labels ALMA_b0 and ALMA_b1 mean the ALMA without and with obstacles simulated, respectively.

4.2.1. Packet loss rate evaluation

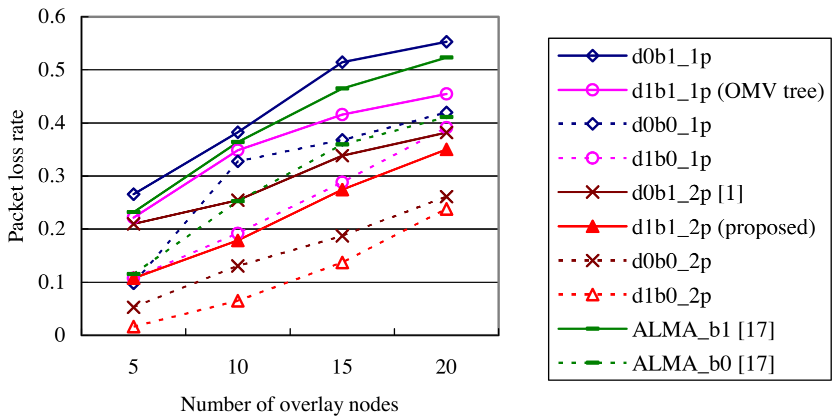
For each overlay case (see Table 4), the packet loss rate with respect to the number of overlay nodes is shown in Fig. 10. We discuss the simulation results in five aspects:

4.2.1.1. The impact of obstacles. Fig. 10 shows that the existence of obstacles significantly affects the stream's packet loss rate. For example, in Fig. 10a, 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 Fig. 10a–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.

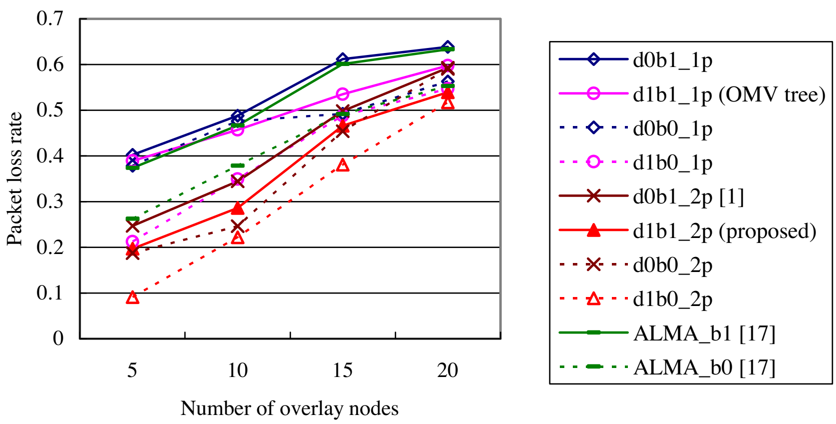
4.2.1.2. The improvement of using dynamic overlays instead of static overlays. To evaluate the impact of using a dynamic overlay, Fig. 10a shows that d1b1_1p (a dynamic tree overlay) is better than d0b1_1p (a static tree overlay) and d1b1_2p (a dynamic mesh overlay) is better than d0b1_2p (a static mesh 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 64–256 kbps, on average. The dynamic overlay significantly improves the static overlay. That is, the dynamic overlay is recommended for urban VANETs. In addition, in Fig. 10a–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 64–256 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.



(a) Stream rate of 64 *kbps*.



(b) Stream rate of 128 *kbps*.



(c) Stream rate of 256 *kbps*.

Fig. 10. Packet loss rate comparison among different cases of overlays: (a) stream rate of 64 kbps; (b) stream rate of 128 kbps; and (c) stream rate of 256 kbps.

4.2.1.3. The improvement of using mesh overlays instead of tree overlays. To evaluate the impact of using mesh overlays, Fig. 10a 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 64–256 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.

4.2.1.4. The impact of number of overlay nodes. For any case of the overlays in Fig. 10a–c, the packet loss rate increases as the number of overlay nodes becomes larger. In addition, in Fig. 10a, 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.

Fig. 10b and c use different stream rates, 128 kbps and 256 kbps, respectively. They show the same observations as Fig. 10a that dynamic overlays are better than static overlays, and two-parent mesh overlays are better than one-parent tree overlays. Fig. 14 shows the impact of a different stream rate to the packet loss. Fig. 14a is the case of a two-parent mesh overlay (d1b1_2p) and Fig. 14b 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. Fig. 11a 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 Fig. 11a 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.

4.3. End-to-end delay evaluation

Fig. 12 compares the stream's end-to-end delay. Non-obstacle cases are not shown here, since they are not real cases in urban VANETs. Fig. 12a–c show that the dynamic

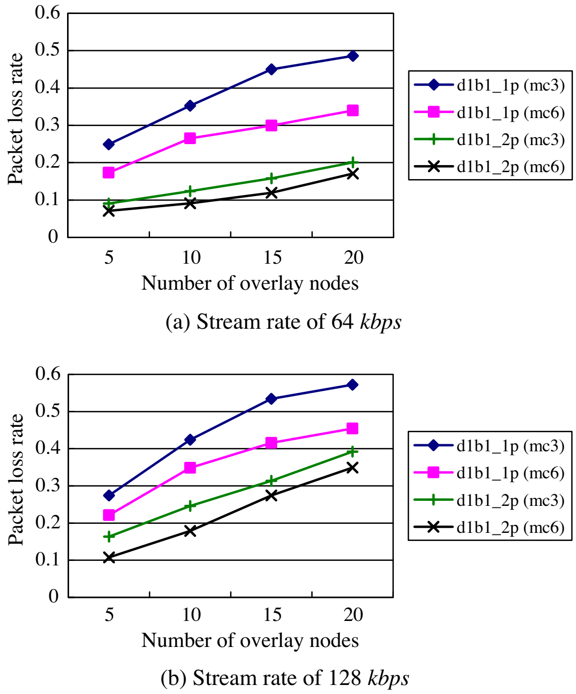
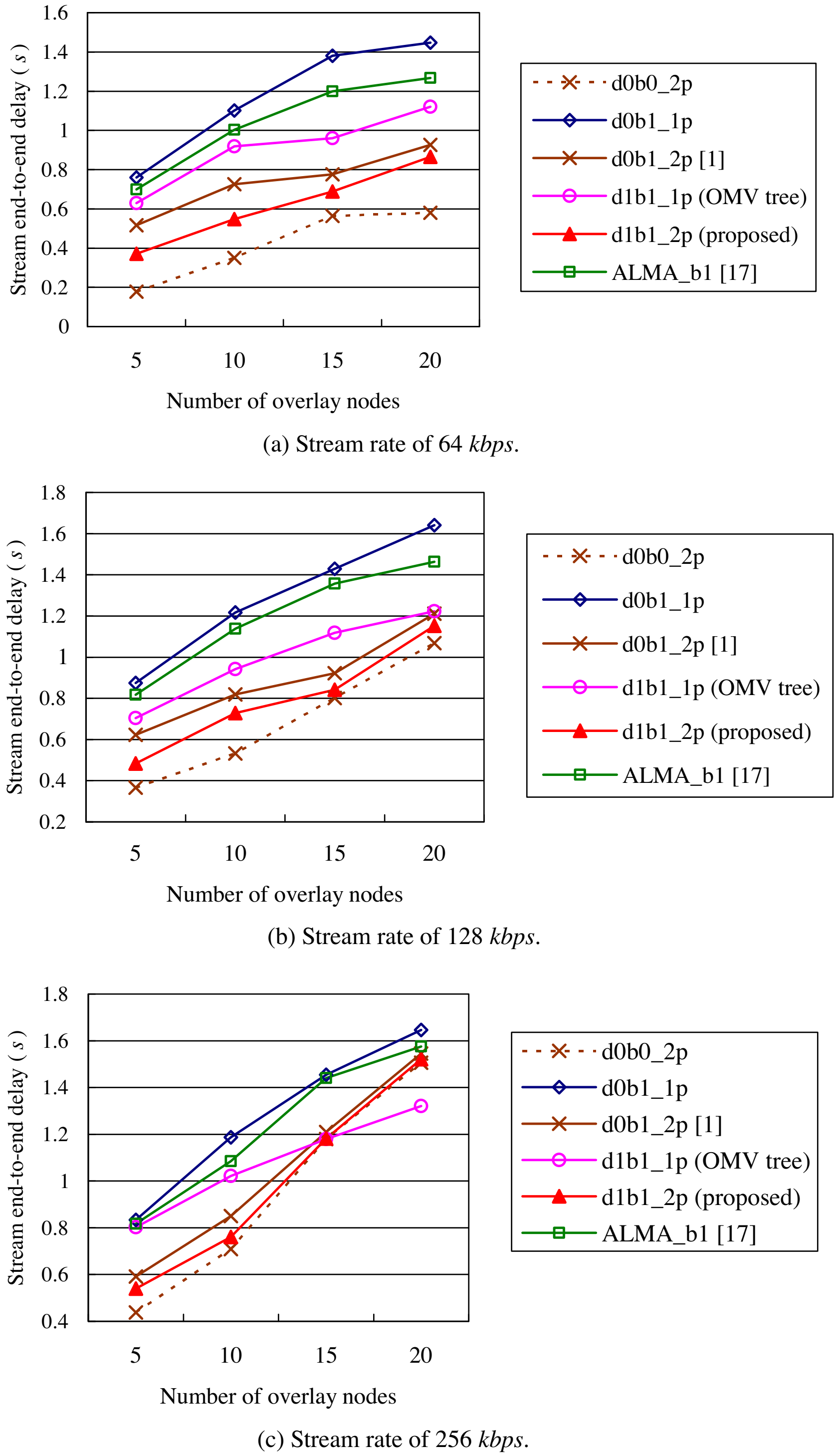


Fig. 11. Packet loss rates of the proposed OMV for tree and mesh overlays using different MAX_CHILDREN: (a) stream rate of 64 kbps; and (b) stream rate of 128 kbps.

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 Fig. 12a–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 64–256 kbps, on average. Comparing the end-to-end delay of the proposed OMV for tree overlays (d1b1_1p) to that of ALMA, in Fig. 12a–c, a decrease of 13.1% is obtained, with group sizes of 5–20 and stream rates of 64–256 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*.

Fig. 12. End-to-end delay comparison among different cases of overlays: (a) stream rate of 64 kbps; (b) stream rate of 128 kbps; and (c) stream rate of 256 kbps.

4.4. Control overhead evaluation

Control overhead comes from the overlay's construction and dynamic adaptation. Fig. 13 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 Fig. 13a, 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 64–256 kbps. As to tree overlays, as shown in Fig. 13b, the results are similar to those of mesh overlays in Fig. 13a. 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 Fig. 13a 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 Fig. 13, 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 5, it shows a feasible combination of road section size, group size, and stream rate. In Table 5, 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 Fig. 13. 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.

4.5. Feasibility evaluation of the proposed OMV

4.5.1. 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 [33–36]. For example, some recent error-resilient video coding techniques were reported to be able to tolerate a packet loss rate of 30% [33–36]. 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 [33–36]. 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 [33–36]. 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

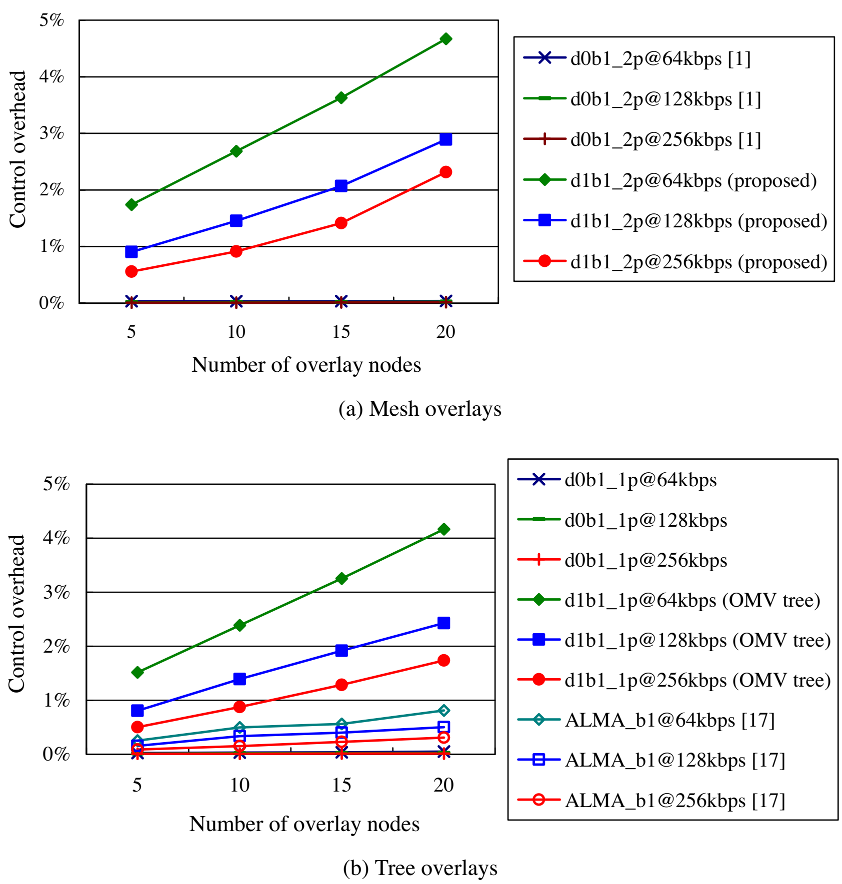


Fig. 13. Control overhead comparison among dynamic and static overlays: (a) mesh overlay; and (b) tree overlay.

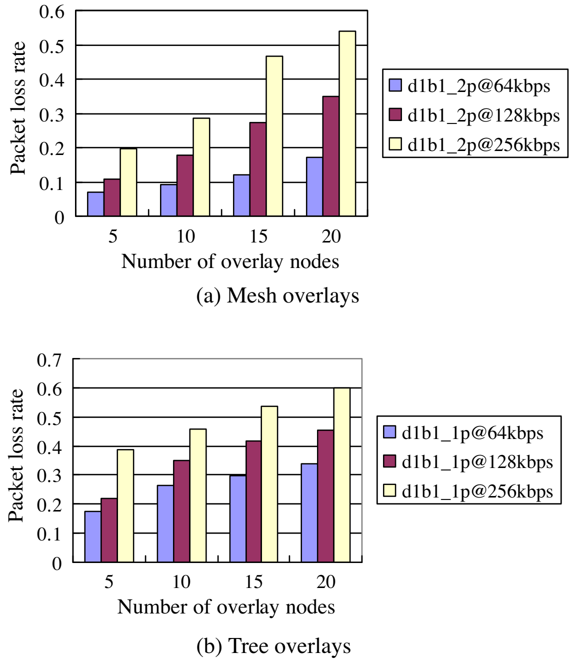


Fig. 14. The impact of a different stream rate to the packet loss rate: (a) dynamic two-parent mesh overlay; and (b) dynamic one-parent tree overlay.

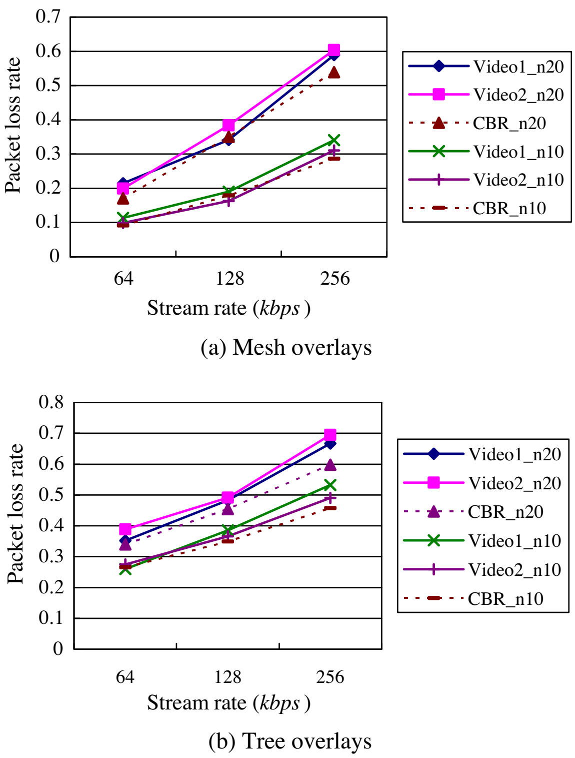


Fig. 15. Packet loss rate comparison among different video clips: (a) dynamic two-parent mesh overlays; and (b) dynamic one-parent tree overlays.

Table 5
Feasible stream rates (kbps) for different road section sizes and overlay sizes.

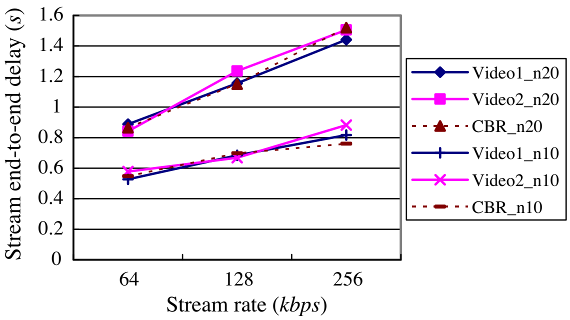
Road section size (m)	Overlay size (Total number of nodes: 100; Field size: 1200 m × 1200 m)			
	5 kbps	10 kbps	15 kbps	20
100	⩽256	⩽128	⩽128	NA
200	⩽256	⩽128	⩽128	⩽64 kbps
400	⩽256	⩽128	⩽64	⩽64 kbps

point out under what configurations to obtain fair video quality. As shown in Fig. 14a, for the two-parent mesh overlay, the packet loss rate is within 0.2 for a 64 kbps stream with 5–20 overlay nodes, and so do the cases for a 128 kbps stream with 5–10 overlay nodes; for a 256 kbps stream with 5–10 overlay nodes, the packet loss rate is within 0.3. As shown in Fig. 14b, 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.

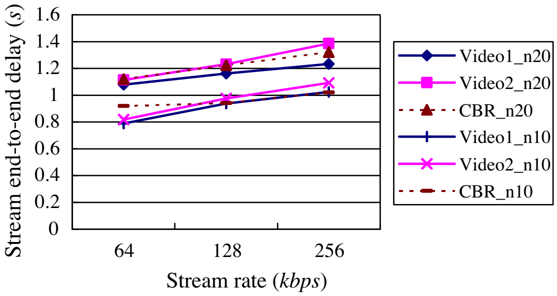
In summary, a two-parent dynamic overlay is feasible for video streaming under the condition of (1) a low stream rate of 64 kbps, or (2) a medium stream rate of 128 kbps with a medium overlay size of 15 overlay nodes, and (3) a high stream rate of 256 kbps with a small overlay size of 10 overlay nodes.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

4.5.2. 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 Figs. 15 and 16. The two real video clips are YUV video sequences in QCIF format (video resolution of 176 × 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/s. The length of *Highway* with 1000 frames is 66.6 s. Since the length of *Carphone* with 150 frames is only 10 s, we replay *Carphone* up to 66.6 s. 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 Figs. 15 and 16, *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. Fig. 15a 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 Fig. 15b, the packet loss rates of the two video clips and



(a) Mesh overlays



(b) Tree overlays

Fig. 16. End-to-end delay comparison among different video clips: (a) dynamic two-parent mesh overlays; and (b) dynamic one-parent tree overlays.

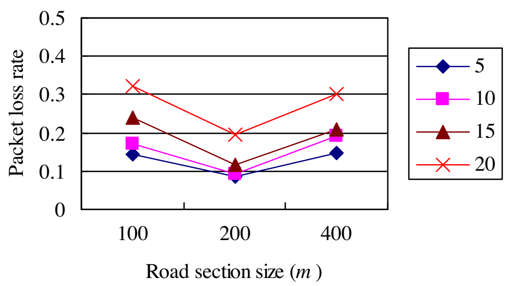
CBR are also close. In addition, Fig. 16a and b show that for the proposed OMV for mesh overlays and the proposed OMV for tree overlays, respectively, under the same stream rate and the same group size, the end-to-end delays of the two video clips and CBR are close. In summary, Figs. 15 and 16 show the feasibility of the proposed OMV for real video.

4.6. The impact of road section sizes to overlays performance

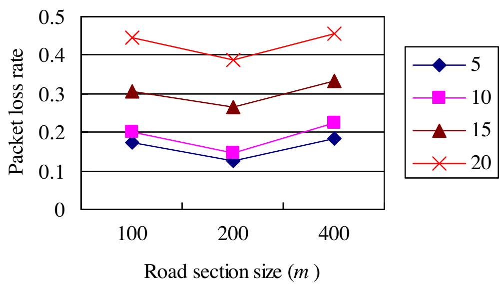
We want to investigate the impact of road section sizes to overlays performance, with various stream rates and overlay sizes. In Figs. 17 and 18, the city map sizes are set as 1200 m × 1200 m, instead of 1000 m × 1000 m, so as to have road section sizes of 100 m, 200 m, and 400 m. The other settings remain the same.

4.6.1. The impact of road section sizes to packet loss rates

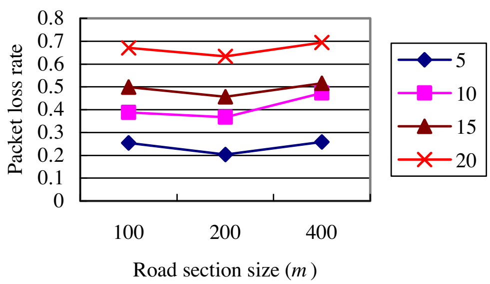
Fig. 17 shows the packet loss rate with road section size of 200 m is lower than that of road section sizes 100 m and 400 m, respectively. With the same number of nodes, when the road section size increases, the number of nodes on each road section becomes larger, which implies higher connectivity, and 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 Fig. 17. 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 Fig. 17.



(a) Stream rate of 64 *kbps*.

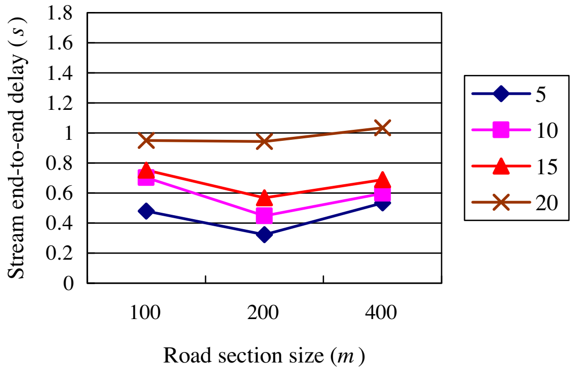


(b) Stream rate of 128 *kbps*.

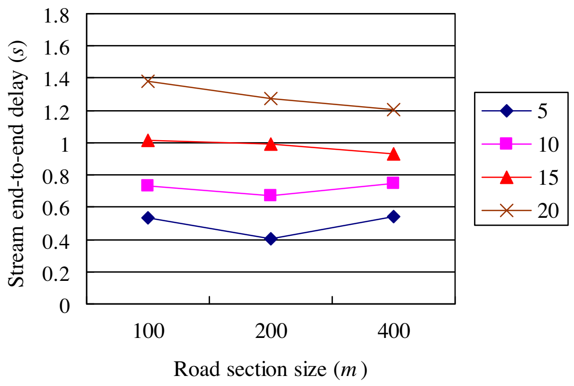


(c) Stream rate of 256 *kbps*.

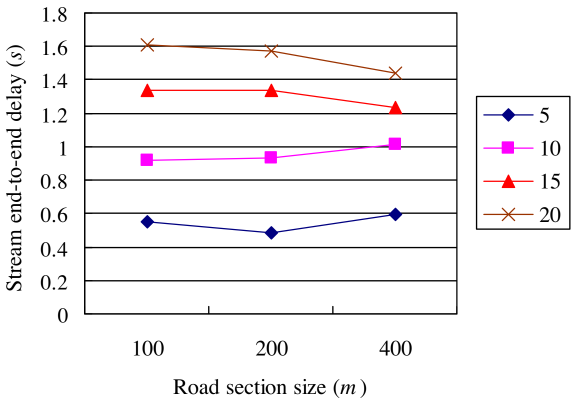
Fig. 17. Packet loss rates under different road section sizes: (a) stream rate of 64 kbps; (b) stream rate of 128 kbps; and (c) stream rate of 256 kbps.



(a) Stream rate of 64 *kbps*.



(b) Stream rate of 128 *kbps*.



(c) Stream rate of 256 *kbps*.

Fig. 18. End-to-end delays under different road section sizes: (a) stream rate of 64 kbps; (b) stream rate of 128 kbps; and (c) stream rate of 256 kbps.

4.6.2. The impact of road section sizes to end-to-end delays

Fig. 18 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 Fig. 17. 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 Fig. 18b 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.

4.6.3. 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 Fig. 17a, 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 Fig. 17b, 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 Fig. 17c, 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 5 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.

4.7. 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.

5. Conclusion

This paper explores 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. We found that the proposed OMV is feasible for video streaming under the following scenarios: (1) a low stream rate of 64 kbps with a large overlay size of 20 overlay nodes, (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. As to city maps of different road section sizes, we have derived feasible stream rates for different overlay 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 higher QoS.

Acknowledgment

The support by the National Science Council under Grants NSC99-2219-E-009-006 and NSC99-2221-E-009-081-MY3 is gratefully acknowledged.

References

[1] N.N. Qadri, M. Fleury, M. Altaf, M. Ghanbari, Multi-source video streaming in a wireless vehicular ad hoc network, IET Communications 4 (11) (2010) 1300–1311.
[2] U. Lee, J.-S. Park, J. Yeh, G. Pau, 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, 2006, pp. 1–5.
[3] J. -S Park, U. Lee, S. Y. Oh, M. Gerla, Emergency related video streaming in VANET using network coding, in: Proc. of the Third International Workshop on Vehicular Ad Hoc Networks, ACM, 2006, pp. 102–103.
[4] M. Abuelela, 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, 2007, pp. 2–8.
[5] Y.-C. Chu, 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, P. Chaparro, Streaming media distribution in VANETs, in: Proc. of the IEEE GLOBECOM, 2008, pp. 1–6.
[7] Q. Li, Y. Andreopoulos, M. van der Schaar, Streaming-viability analysis and packet scheduling for video over in-vehicle wireless networks, IEEE Transactions on Vehicular Technology 56 (2007) 3533–3549.
[8] M. Guo, M.H. Ammar, 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, 2005, pp. 171–180.
[9] Y. Ko, N. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, Wireless Networks 6 (2000) 307–321.
[10] M. Fiore, J. Harri, F. Filali, C. Bonnet, Vehicular mobility simulation for VANETs, in: Proc. of the 40th Annual Simulation Symposium, 2007, pp. 301–309.
[11] N.N. Qadri, M. Altaf, M. Fleury, M. Ghanbari, 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, 2009, pp. 429–434.
[12] J. Nzouonta, N. Rajgure, G. Wang, C. Borcea, VANET routing on city roads using real-time vehicular traffic information, IEEE Transactions on Vehicular Technology 58 (7) (2009) 3609–3626.
[13] Y.H. Ho, A.H. Ho, K.A. Hua, Routing protocols for inter-vehicular networks: a comparative study in high-mobility and large obstacles environments, Computer Communications 31 (2008) 2767–2780.
[14] J. Zhao and G. Cao, VADD: vehicle-assisted data delivery in vehicular ad hoc networks, in: Proc. of the 25th IEEE INFOCOM, 2006, pp. 1–12.
[15] T. Rasheed, M. Jerbi, S.M. Senouci, Y. Ghamri-Doudane, Towards efficient geographic routing in urban vehicular networks, IEEE Transactions on Vehicular Technology 58 (9) (2009) 5048–5059.
[16] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: Proc. of the IEEE WCNC, 2003, pp.1118–1123.
[17] M. Ge, S.V. Krishnamurthy, M. Faloutsos, Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol, Ad Hoc Networks 4 (2) (2006) 283–300.
[18] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: Proc. of the IEEE WCNC, 2003, pp. 1118–1123.
[19] K. Kim, S.-H. Kim, A novel overlay multicast protocol in mobile ad hoc networks: design and evaluation, IEEE Transactions on Vehicular Technology 52 (6) (2005).
[20] Qualnet, http://www.scalable-networks.com/.
[21] D. Jiang, L. Delgrossi, IEEE 802.11p: towards an international standard for wireless access in vehicular environments, in: Proc. of the IEEE Vehicular Technology Conference, 2008, pp. 2036–2040.
[22] Z. Yang, M. Li, W. Lou, CodePlay: live multimedia streaming in VANETs using symbol-level network coding, in: Proc. of the 18th IEEE International Conference on Network Protocols, 2010, pp. 223–232.
[23] S.-Y. Wang, P.-F. Wang, Y.-W. Li, 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), 2011, pp. 1937–1942.
[24] N.N. Qadri, M. Fleury, M. Altaf, 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, 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, 1996, p. 910.

[27] C.-O. Chow, H. Ishii, Enhancing real-time video streaming over mobile ad hoc networks using multipoint-to-point communication, Computer Communication 30 (2007) 1754–1764.
[28] J. He, A. Chaintreau, C. Diot, A performance evaluation of scalable live video streaming with nano data centers, Computer Networks 53 (2) (2009) 153–167.
[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 5 (4) (2011) 440–450.
[31] F.J. Martinez, C.-K. Toh, J.-C. Cano, C.T. Calafate, P. Manzoni, Realistic radio propagation models (RPMs) for VANET simulation, in: Proc. of the IEEE Wireless Communications and Networking Conference, 2009, pp. 1155–1160.
[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, A. Pitsillides, Atherosclerotic plaque ultrasound video encoding, wireless transmission, and quality assessment using H.264, IEEE Transactions on Information Technology in Biomedicine 15 (3) (2011).
[34] T. T. Thai, J. Lacan, H. Meric, Error tolerance schemes for H.264/AVC: an evaluation, in: Proc. of IEEE Consumer Communications & Networking Conference (CCNC), 2012, pp. 571–575.
[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), 2011, pp. 1057–1060.
[36] F. Zhu, W. Zhang, N. Yu, J. Xu, 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, 2011, pp. 609–612.
[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, 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.

Yi-Ling Hsieh received the B.S. and M.S. degrees in computer and information science from the National Chiao Tung University, Taiwan, in 2003 and 2005, respectively. She is currently a Ph.D. student in the Department of Computer Science, National Chiao Tung University. Her research interests include wireless (ad hoc/sensor/VANET) networks, mobile computing, and efficient energy management for portable wireless/multimedia devices.

Kuochen Wang received the B.S. degree in control engineering from the National Chiao Tung University, Taiwan, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from the University of Arizona in 1986 and 1991, respectively. He is currently a Professor in the Department of Computer Science, National Chiao Tung University. He was a Director in the Institute of Computer Science and Engineering/Institute of Network Engineering, National Chiao Tung University from August 2009 to July 2011. He was an Acting/Deputy Director of the Computer and Network Center at this university from June 2007 to July 2009. He was a Visiting Scholar in the Department of Electrical Engineering, University of Washington from July 2001 to February 2002. From 1980 to 1984, he was a Senior Engineer at the Directorate General of Telecommunications in Taiwan. He served in the army as a second lieutenant communication platoon leader from 1978 to 1980. His research interests include wireless (ad hoc/sensor/VANET) networks, mobile/cloud computing, and power management for multimedia portable devices.