Journal of Information Processing Vol. 18 110–124 (Mar. 2010)

Regular Paper

A Novel Multi-hop Broadcast Protocol for Vehicular Safety Applications

Celimuge Wu,[†1] Kazuya Kumekawa[†1] and Toshihiko Kato[†1]

Many safety applications in Vehicular Ad hoc Networks (VANETs) are based on broadcast. Designing a broadcast protocol that satisfies VANET applications' requirements is very crucial. In this paper, we propose a reliable and efficient multi-hop broadcast routing protocol for VANETs. The proposed protocol provides the strict reliability in various traffic conditions. This protocol also performs low overhead by means of reducing rebroadcast redundancy in a high-density network environment. We also propose an enhanced multipoint relay (*MPR*) selection algorithm that considers vehicles' mobility and then use it for relay node selection. We show the performance analysis of the proposed protocol by simulation with ns-2 in different conditions, and give the simulation results demonstrating effectiveness of the proposed protocol compared with other VANET broadcast schemes.

1. Introduction

A Vehicular Ad hoc Network (VANET) is a form of mobile ad hoc networks providing communications among nearby vehicles, and between vehicles and nearby fixed roadside equipment. The opportunities for VANETs are growing rapidly. Many VANET applications are broadcast based and thus multi-hop broadcast is required to disseminate information to desired receivers. The simplest way to disseminate information is flooding. However, flooding has serious problems. First of all, simple flooding cannot provide enough reliability. Its non-support of retransmission degrades the data delivery rate and its redundant rebroadcasts described below causes many collisions resulting in making the delivery rate worse. In the heavy traffic condition where the VANET communication is likely to be exploit, many vehicles exist in a dense manner within a radio transmission range. In such a high-density network environment, flooding introduces redundant rebroadcast, that is, many vehicles within a radio transmission range of one vehicle try to rebroadcast a received message, and it causes high overhead in the data dissemination.

Although there are many proposals on VANET broadcast protocols focusing on the reliability and the efficiency in a vehicular environment, they have some limitations. Proposals focusing on the reliability do not consider high-density environments. On the other hand, those focusing on the efficiency in high-density environments are only designed for dense networks and provide poor performance in the sparse network environment.

In this paper, we first propose a relay node selection algorithm (enhanced *MPR* selection algorithm) considering network mobility. Based on it, we then propose a reliable and efficient broadcast protocol that can work well in various traffic conditions. The proposed protocol uses a hop-by-hop retransmission scheme to provide the strict reliability in various traffic conditions. This protocol also provides low overhead in a high-density network environment by means of introducing boundary nodes which are in charge of rebroadcast. Our protocol also works well in a sparse network. We confirm the effectiveness of the proposed protocol through simulations using the ns-2 network simulator [1].

The remainder of the paper is organized as follows; In Section 2, we give a brief description of the related work and elaborate our contribution. We propose an enhanced *MPR* selection algorithm in Section 3. Next in Section 4, we give the detailed description of the proposed protocol. We evaluate the protocol's performance in Section 5. Finally, the conclusions are presented in Section 6.

2. Related Works and Our Contribution

2.1 Related Works on Broadcasting in VANETs

The main role of broadcast protocols in VANETs is to disseminate safety messages. Therefore, many researchers aim to improve reliability of VANETs. Tonguz, et al. [2] propose a distributed vehicular broadcasting protocol designed for safety and transport efficiency applications in VANETs. Liu, et al. [3] analyze and evaluate techniques for achieving reliable broadcast in error-prone multi-hop wireless networks, and propose an overall algorithm encompassing a combination

†1 Graduate School of Information Systems, University of Electro-Communications

of the investigated techniques as an efficient solution for reliable broadcasting in multi-hop wireless networks. Jiang, et al. [4] propose an alarm message broadcast routing protocol REAR, which has higher reliability than a location-based algorithm with fewer broadcast packets. Khakbaz, et al. [5] present a method that improves the delivery rate of broadcast messages by overcoming problem of connectivity gaps by sending small messages periodically. However, Refs. 3)–5) do not consider high-density network environments at all. Besides, the proposals in Ref. 2) and Ref. 3) do not consider topology changes caused by vehicles' movements. Ref. 4) suffers from a higher dissemination latency.

Other researchers focus on efficient broadcast methods for VANETs in the high-density environments. Ref. 6) proposes three probabilistic and timer-based broadcast suppression techniques. Ref. 7) presents an opportunistic routing protocol that uses a modified 802.11 MAC layer using active signaling to select the best relay from all the vehicles that have correctly received the packet. Since both of those proposals do not introduce strict data delivery schemes, it is possible that they cannot work well in sparse network environments or under medium or low traffic load conditions.

In high-density networks, it is possible to reduce broadcast redundancy by selecting a small subset of nodes to relay a data packet. To efficiently broadcast messages in vehicular ad hoc networks, relay node selection should be handled efficiently. Many methods to select relay nodes are proposed [8)–11)]. However, none of them considers nodes' mobility in relay node selection. Therefore, they are not suitable for highly dynamic vehicular ad hoc networks.

2.2 Our Contribution

The broadcast routing protocols in VANETs should provide high delivery rate and should be lightweight and suitable for different traffic conditions. In this paper, we first propose an enhanced *MPR* selection algorithm considering network mobility. Based on the algorithm, we then propose a multi-hop broadcast protocol that can deliver safety messages to all desired receivers. The proposed protocol uses selected boundary nodes to relay data avoiding broadcast storm problems in high-density networks. The boundary node rebroadcast mechanism substantially reduces the message overhead as compared to a simple flooding mechanism. The proposed protocol is robust to mobility and channel error by use of a strict retransmission mechanism in case of packet losses.

3. Enhanced *MPR* Selection Algorithm

In order to reduce redundant broadcast in high-density networks, the messages should be only rebroadcast by a subset of neighbors. Without loss of generality, we use two-hop neighbor information to select relay nodes. We assume every node broadcast hello messages periodically. Every vehicle places its one-hop neighbor information to hello messages and therefore vehicles are aware of their two-hop neighbors. We do not assume a GPS like positioning device is available for every vehicle. (The terms node and vehicle are used interchangeable in this paper.)

3.1 Problems in the Original *MPR* Selection Algorithm

Although the original *MPR* selection algorithm [8] based broadcast scheme could efficiently reduce redundant rebroadcast in static networks [12], it may fail in dynamic networks. We use **Fig. 1** as an example. In figures we use in this paper, $TR(x)$ shows the transmission range of node x. As shown in Fig. 1 (a1), S receives hello from neighbors and updates two-hop neighbor information. The network topology changes to a new state, which is shown in Fig. 1 (a2). S intends to send data at this time and select B2 as a relay node due to its out-of-date two-hop neighbor information. Obviously, B2 is no longer the node that provides maximal additional coverage. We use additional coverage of node x, $AC(x)$, to mean the set of nodes which are one-hop neighbors of the node x but not one-hop neighbors of the sender node s. Specifically, $AC(x)$ is defined as

$$AC(x) = \overline{N(s)} \cap N(x), \tag{1}$$

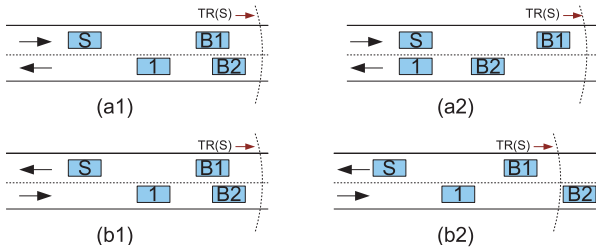

Fig. 1 Cases of *MPR* selection failure.

where $N(x)$ and $N(s)$ denote a one-hop neighbor set of node x and one-hop neighbor set of sender node s respectively. We note here that node x belongs to $N(s)$.

The original *MPR* selection algorithm also fails in case of another situation, which is shown in Fig. 1 (b1, b2). S updates neighbor information when B2 is the best relay node as Fig. 1 (b1) shows. Network topology changes to a new state (Fig. 1 (b2)) and S selects node B2 as relay node based on previous knowledge. As a consequence, B2 could not receive the data packet and the data could not be transmitted to two-hop neighbors.

In short, out-of-date neighbor information influences effectiveness of the original *MPR* selection algorithm. Obviously, it is important to consider node mobility in *MPR* selection in vehicular ad hoc networks. Here we propose an enhanced *MPR* selection algorithm considering network mobility.

3.2 Enhancement of *MPR* Selection Considering Network Mobility

3.2.1 Notation

Before we proceed further, we summarize the notation we use in this paper (shown in **Table 1**).

3.2.2 *MPR* Selection Criteria

The original *MPR* selection algorithm [8] considers additional coverage (as shown in Eq. (1)) only. Of course, additional coverage is an important factor, but not all. In this paper, we define predicted *MPR* fitness (*PMF*) to evaluate a node whether it is suitable for relaying data packet or not. To calculate $PMF(x)$ for node x, we first introduce multipoint relay fitness ($MF(x)$) as

$$MF_i(x) = \frac{|AC_i(x)|}{|N_i(s) \cup N_i(x)|} \tag{2}$$

where i indicates the current value.

When a node s receives hello message from node x, it calculates corresponding $MF_i(x)$. In Eq. (2), $N_i(x)$ denotes neighbor set of node x, $|N_i(x)|$ denotes number of x's one hop neighbors. Eq. (2) could give higher value to nodes that have larger additional coverage. However, it is not sufficient to only consider the additional coverage in dynamic networks. So, we consider nodes' movement in *PMF* calculation. In order to provide different weights to different levels of movement, we include discount rate γ.

Table 1 Notation.

MF	multipoint relay fitness
PMF	predicted MPR fitness
$PMF_i(x)$	current PMF of node x
$PMF_{i-1}(x)$	previous PMF of node x
$AC(x)$	additional coverage of node x
$\|A\|$	number of elements in set A
$ACN(x)$	$\|AC(x)\|$, number of elements in $AC(x)$
$ACN_{\min}$	minimal ACN between one-hop neighbors
$ACN_{\max}$	maximal ACN between one-hop neighbors
ACN_{Thresh}	a threshold value which is used to determine MPR candidate nodes
$N(x)$	one-hop neighbor set of node x
$N(s)$	one-hop neighbor set of sender node s
$N^2(s)$	two-hop neighbor set of sender node s
α	a rate which denotes how much current value contribute to the new value
γ	discount factor
$MPR(s)$	multipoint relay set of sender node s

$$\gamma = \begin{cases} \sqrt{\frac{|AC_i(x) \cap AC_{i-1}(x)|}{|AC_i(x) \cup AC_{i-1}(x)|}}, & \text{if } AC_i(x) \cup AC_{i-1}(x) \neq \phi \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $i-1$ indicates the previous value. Eq. (3) could give a larger value to same directed vehicles and smaller value to vehicles moving in the opposite direction. If a node x is moving in the opposite direction to the sender, the corresponding γ will be smaller than other vehicles moving in the same direction because its additional coverage is frequently changing.

We also consider nodes' history in *PMF* calculation. Because, in general, if a link's duration time is long, it will be more likely to be durable in the future. For example, sender should use vehicles in same lane or same direction to forward data. We use α to consider node's history in *PMF* calculation. α is a rate that denotes how much current value contributes to the new value.

Considering node's current state, history and movement, we update a neighbor's *PMF* as follows.

$$PMF_i(x) \leftarrow (1-\alpha)PMF_{i-1}(x) + \alpha \times \gamma \times MF_i(x). \tag{4}$$

$PMF_i(x)$ is updated upon reception of a hello from a neighbor. Every node maintains a PMF ($PMF_{i-1}(x)$) and a AC ($AC_{i-1}(x)$) for every one-hop neighbor. In Eq. (4), if it is the first PMF calculation, the $PMF_{i-1}(x)$ will be set to 0. Similarly, as far as Eq. (3) is concerned, if it is the first AC calculation, the $AC_{i-1}(x)$ will be set to ϕ. The sender node uses these values and the current MF ($MF_i(x)$) and AC ($AC_i(x)$) to calculate the latest PMF ($PMF_i(x)$) as shown in Eq. (3) and Eq. (4). The node then updates the $PMF_{i-1}(x)$ and $AC_{i-1}(x)$ it maintains. In the proposed algorithm, $PMF(x)$ will be reset to zero if the sender did not hear from x in three times the hello interval.

Note that α is a design parameter that should be carefully chosen. If the value is too small, PMF will not adapt quickly to network dynamics. A higher α discounts older observations faster. However, if the value is too large, then the PMF cannot reflect network movement tendency because it will be vulnerable to temporary misleading values. Through simulations, we observe that 0.6 to 0.8 are better values for α. However, we can not see significant differences between them. Therefore, we set α to 0.7.

3.2.3 *MPR* Selection Procedure

Senders (broadcast source nodes or relay nodes) in vehicular ad hoc networks could be divided to the following two different types, according to their broadcast intentions.

(1) There is one type of senders that only need to disseminate messages in one direction. In general, relay nodes (except nodes near an intersection) belong to this type. These senders use algorithm 1, which will be described later. Here we use intersection to mean a road junction where two or more roads either meet or cross at the same level.

(2) There also exists another type of sender that require disseminating messages in more than one direction. Broadcast source nodes always need to select at least two relay nodes to guarantee dissemination of messages in both forward and backward directions. Senders that are near to an intersection also need to disseminate messages in more than one direction. These senders use algorithm 2, which will be described later. In this paper, we assume vehicles know they are near an intersection or not. This can be achieved by beaconing of access point at the intersection.

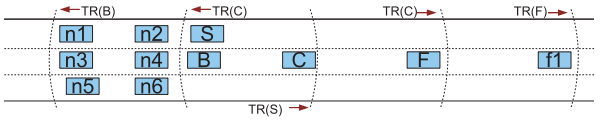

Fig. 2 An example for algorithm 1.

Algorithm 1: The sender first calculates a threshold value ACN_{Thresh} as

$$ACN_{Thresh} = ACN_{\min} + (1-\beta) \times (ACN_{\max} - ACN_{\min}), \quad (5)$$

where ACN denotes the number of elements in AC. $ACN_{\min}$ is the minimal ACN between neighbors in the forward direction and $ACN_{\max}$ is the maximal ACN between neighbors in the forward direction. We use neighbors in the forward direction to mean the neighbor nodes that are not neighbors of the upstream node. The sender node can get its neighbors in the forward direction by simply excluding the upstream node's neighbors from one-hop neighbors. From the neighbors in the forward direction, the sender node first selects the nodes that have larger ACN than ACN_{Thresh} as MPR candidates. The sender then specifies the node that has maximal PMF between these MPR candidates as the relay node.

Here we use **Fig. 2** to explain why the proposed protocol selects MPR candidates from neighbors in the forward direction. We assume node S is a sender node and it specifies node C as a boundary node and broadcasts a data packet. Upon reception of the data packet, node C specifies the next relay node. Obviously, we know that node C should select node F as MPR candidates because it has to disseminate information to node f1. However, if we select MPR candidates from all one-hop neighbors, node B will be selected because node B's ACN is much larger than node F's. Fortunately, in algorithm 1, because MPR candidates are selected from the neighbors in the forward direction, node C can select node F as MPR candidates.

In algorithm 1, the value of β determines the set of MPR candidates. If the value is 0, ACN_{Thresh} will be $ACN_{\max}$. Thus only the nodes that have maximal additional coverage will be selected as MPR candidates. If the value is 1, the ACN_{Thresh} will be $ACN_{\min}$. In that case, the set of MPR candidates will be its neighbors in the forward direction. It means that the sender node will select the relay nodes totally based on PMFs of its neighbors in the forward direction. As a

result, the sender node may select a node that is very near, resulting in inefficient relay. So, we use β to control the value of threshold. Through simulations, we know that generally, selecting the first quarter of nodes according to the values of ACN results in good performance outcome in various node densities. So we set β to be 1/4.

Algorithm 2:

(1) Start with an empty multipoint relay set $MPR(s)$ where s indicates the sender node.

(2) First select those one-hop neighbor nodes in $N(s)$ as multipoint relays which are the only neighbor of some node in two-hop neighbor set ($N^2(s)$), and add these one-hop neighbor nodes to the multipoint relay set $MPR(s)$.

(3) While there still exist some node in $N^2(s)$, which is not covered by the multipoint relay set $MPR(s)$:

(a) For each node in $N(s)$, which is not in $MPR(s)$, compute the number of nodes that it covers among the uncovered nodes in set $N^2(s)$.

(b) Add that node of $N(s)$ in $MPR(s)$ for which this number is maximum. If more than one node has the same number, we pick the node which has maximal PMF.

As described above, a sender uses algorithm 1 or algorithm 2 depending on its current state. If the node only needs to disseminate information in one direction, it uses algorithm 1 and otherwise uses algorithm 2. The enhanced *MPR* selection algorithm evaluates nodes' *MPR* fitness based on two-hop neighbor information. The algorithm considers nodes' history and moving tendency in the *MPR* selection procedure therefore can use better nodes to relay messages. By selecting the relatively stable nodes, the algorithm also increases the probability of disseminating more than one packet using same relay node. This feature could help broadcast protocol to reduce acknowledgement messages while ensuring reliability. The proposed broadcast protocol that uses this algorithm will be explained in the next section.

3.3 Effectiveness of the Proposed Algorithm

Due to dynamic features of VANETs, the original *MPR* selection algorithm [8] did not work well. To solve this problem, the enhanced *MPR* selection algorithm picks relay node considering mobility. Different to other mobile ad hoc networks, relay node selection in VANETs is relatively straightforward. Because lane width is much smaller than transmission range, a sender always needs to select only one forwarder in one direction. Taking advantage of this feature, algorithm 1 selects the best relay node.

In case of senders need to disseminate messages in different direction, algorithm 2 can enhance the original *MPR* algorithm using mobility awareness. In that case, multiple nodes may have similar additional coverage, so choosing the best one is particularly important. However, the original *MPR* algorithm may select any of them. If the selected nodes have the opposite direction to the sender, it would result in low dissemination speed or dissemination failure as described above. Algorithm 2 enhances the original *MPR* algorithm when there are multiple candidate nodes that have same additional coverage range. Since considering network mobility, the proposed algorithm ensures selecting relatively stable nodes to forward data. In general, the proposed algorithm could eliminate errors of the original *MPR* algorithm which occurs from imprecise topology information. In the worst case, the proposed protocol performs same as the original *MPR* algorithm.

4. Protocol Design

4.1 Design Principles

We propose here a multi-hop broadcast protocol which uses enhanced *MPR* selection algorithm proposed above. This protocol aims to ensure the strict reliability as well as the transmission overhead minimization. As for the strict reliability, we use the following scheme. We use hop-by-hop manner to provide reliability. Every sender is responsible for assuring reliable broadcast to its one-hop down stream nodes. A sender includes a TO-ACK-LIST in a data packet, and the nodes included in the TO-ACK-LIST will reply ACK to the sender when it receives the packet. We use three types of acknowledgement methods (explicit ACK, implicit ACK and negative ACK). While broadcasting a data packet, a sender starts a retransmission timer. A sender node maintains a TO-ACK-LIST locally to store nodes from which it has not heard ACK. It removes the corresponding node from the list upon reception of an ACK. If the local TO-ACK-LIST is not NULL when the retransmission timer expires, the sender

will retransmit the packet.

In order to reduce rebroadcast redundancy in high-density networks, the proposed protocol uses only a subset of nodes in the network to relay received broadcast packets. We assume vehicles exchange their neighbor information through hello messages. Every vehicle places its neighbor information to hello message and therefore vehicles know existence of their two-hop neighbors. Before broadcasting a packet, a sender uses the enhanced *MPR* selection algorithm to decide relay nodes based on two-hop neighbor information. We describe these relay nodes as boundary nodes. A sender includes the list of its boundary nodes (BOUNDARY-LIST) in a data packet. Upon receiving a broadcast packet, the nodes will rebroadcast the packet if they are included in the BOUNDARY-LIST.

In order to cope with the network topology change, the information, REVERSE-BOUNDARY-LIST, is attached to a data packet. Before sending a broadcast packet, a sender will append their address to the REVERSE-BOUNDARY-LIST. Therefore, REVERSE-BOUNDARY-LIST of a packet is composed of the addresses of the nodes that have forwarded the packet. Every node also needs to maintain a unicast route table, which is used to send ACK. Upon receiving a packet, every node maintains route entries to nodes contained in the REVERSE-BOUNDARY-LIST. These routes use the sender node (the last node relayed the packet) as the next hop. They are used to deliver ACK to two-hop upstream sender in case the topology changes. The protocol uses these routes for mobility handling in a manner we will explain later.

The proposed protocol intends to use relatively far nodes to relay packets because they can provide larger progress on distance. We have to note here that bit error rates in 802.11a/b/g/p are relatively high between far nodes than near nodes. However, in this paper, we assume bit error rates are unaffected by distance between sender node and relay node.

4.2 Protocol Information and Acknowledgment Scheme

Every sender node maintains a broadcast cache, which consists of entries that include the following fields.

- **Source node address and broadcast ID**
- **TO-ACK-LIST:** This list consists of nodes that should acknowledge upon reception of the corresponding packet.
- **Expire time:** The time of the corresponding packet should be retransmitted in case the packet is not successfully received by all desired receivers.
- **Corresponding data packet:** A copy of the data packet that can be used to retransmit.

A data packet includes the following fields in addition to data itself.

- **Source node address and broadcast ID**
- **BOUNDARY-LIST:** A list consisting of boundary nodes.
- **TO-ACK-LIST**
- **REVERSE-BOUNDARY-LIST:** A list consisting of nodes which have rebroadcasted this packet.
- **Consecutive broadcasting flag:** A flag shows whether this packet belongs to a consecutive broadcast or not.
- **Retransmit flag and retransmit source node address:** Retransmit flag shows whether this packet is a retransmitted packet or not. Retransmit source node address is the address of the node which initiates retransmission.

As mentioned above, we use the following three types of acknowledgement methods.

- **Explicit ACK:** An explicit ACK should include source node address, broadcast ID, and receiver's address (address of ACK sender).
- **Implicit ACK:** A rebroadcast packet is an implicit ACK to the sender's upstream node. Upon hearing the packet, the upstream node knows the packet have been successfully received by the downstream node.
- **Negative ACK (NACK):** A negative ACK should include all fields of explicit ACK. Upon reception of a NACK, the sender rebroadcasts the packet immediately.

4.3 Boundary Specification and TO-ACK-LIST Selection

Our proposed protocol always selects nodes that provide larger progress on distance as boundary nodes. We select boundary nodes using Enhanced Multipoint Relay selection algorithm proposed in Section 3. Every node specifies boundary nodes before broadcasting a message. In this way, redundant broadcasting can be efficiently reduced. If the sender is not the broadcast source, the BOUNDARY-LIST should not include the upstream node's boundary nodes and nodes which are included in REVERSE-BOUNDARY-LIST.

The broadcast source node's TO-ACK-LIST is simply defined as its one-hop neighbors. If the sender is not the broadcast source, its TO-ACK-LIST will exclude nodes that included in the upstream node's TO-ACK-LIST. The TO-ACK-LIST also excludes nodes that included in the packet's REVERSE-BOUNDARY-LIST.

4.4 Packet Rebroadcasting

Before broadcasting a packet, the source node does the following actions:

(1) Update the broadcast cache. Set the expire time field according to delay constraint. Calculate an TO-ACK-LIST for two purposes: Firstly, to maintain locally for future retransmission checking. Secondly, to let downstream nodes know whether the packet should be acknowledged or not. Place TO-ACK-LIST to the data packet.
(2) Select boundary nodes and place them to the data packet.
(3) Place own address to the REVERSE-BOUNDARY-LIST.

Upon receiving a broadcast packet, an intermediate node does following actions:

(1) Create an reverse route to nodes included in the REVERSE-BOUNDARY-LIST for delivering ACK.
(2) **If** (the BOUNDARY-LIST contains the node) **then** {
Update the packet's BOUNDARY-LIST according to own neighbor information.
Append own address to REVERSE-BOUNDARY-LIST.
Update the broadcast cache. Set the expire time field according to delay constrain. Update the TO-ACK-LIST.
Rebroadcast. (Rebroadcast is implicit ACK to the upstream node.)
}**else**{
If (TO-ACK-LIST contains the node) **then** {
Send an ACK to upstream node.
}**else** {
May send multi-hop ACK to nodes included in REVERSE-BOUNDARY-LIST. (This will be explained later in Section 4.6.)
}
}

Upon receiving an ACK (or an implicit ACK), a node does the following actions:

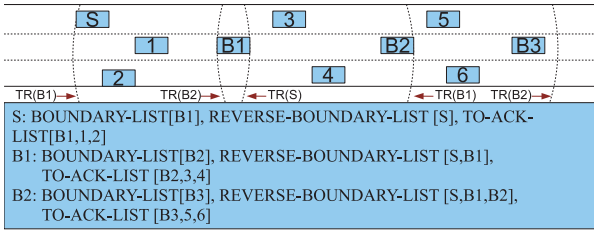

Fig. 3 Boundary specification.

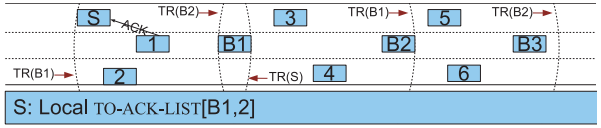

Fig. 4 ACK management.

(1) According to the ACK's information, get the corresponding entry from broadcast cache.
(2) Remove the corresponding node (the sender of the ACK) from local TO-ACK-LIST. **If** the local TO-ACK-LIST is NULL **then** remove the corresponding entry from broadcast cache.

As shown in **Fig. 3**, node S selects node B1 as boundary node and sets TO-ACK-LIST to [B1, 1, 2]. Node S also appends own address to REVERSE-BOUNDARY-LIST. S broadcasts the message and B1 knows itself is a boundary and then updates the packet's boundary nodes to [B2, 3, 4]. Similarly, B1 appends own address to REVERSE-BOUNDARY-LIST before relaying. When node 1 receives the message from S, it sends ACK to S that can be seen in **Fig. 4**. But node 1 does not send ACK to B1 when it receives the message from B1 because it is not specified to do so. Upon reception of the ACK from node 1, S will delete node 1 from local TO-ACK-LIST. As shown in **Fig. 5**, if node S does not receive ACK from node 2 before retransmission timer expires, node S will retransmit the message.

4.5 Retransmit Handing

Every node maintains a retransmission timer and performs retransmission check

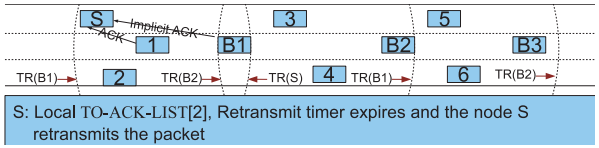

Fig. 5 Retransmission.

periodically. When retransmission timer expires following action is executed.

(1) **If** (exist expired broadcast cache entry) **then**{

Get the corresponding packet and update the packet's retransmit source node address with its address.

Set retransmission flag to 1 and BOUNDARY-LIST to NULL.

Update the packet's TO-ACK-LIST according to local TO-ACK-LIST.

Set the TTL of the packet to 2. (We use two-hop flooding. In general, two-hop flooding is large enough. In case of still have missing receivers, increase TTL by 2 and retransmit.)

Retransmit.

}

Upon receiving a retransmitted data, a node checks if its own address is included in the TO-ACK-LIST of the packet. If so the node sends ACK to the retransmission source node. Otherwise, the node just rebroadcasts the packet.

4.6 Mobility Robustness

We also use ACK messages to handle topology changes. ACK could be one hop or multi-hop. As shown in **Fig. 6** (a), we assume L1 did not receive data from S1 and has moved to new position, which is out of the transmission range of S1. Upon receiving the data from B1, checking the reverse boundary node list, L1 knows the packet has been broadcasted by S1. Because S1 is a neighbor of L1 (in the L1's knowledge), L1 should have received the packet before, but L1 did not receive the packet. This implicates that some link changes may have happened. Therefore, L1 sends ACK to node S1 although not specified by B1 to acknowledge. The ACK message could arrive at node S1 by the way of B1 and then S1 would know L1 is no longer a neighbor.

In case of another situation, which is shown in Fig. 6 (b), L2 received data from S2 and sent ACK back to S2. The ACK is lost and S2 retransmits the

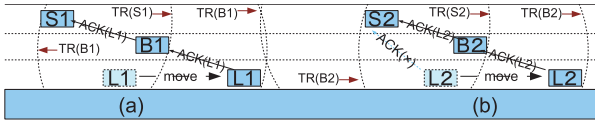

Fig. 6 Mobility robustness.

Table 2 Packets in case of 3 consecutive broadcasting.

packet	consecutive flag	seq no	next packet arrival time
First	1	1	0.5 s
Second	1	2	0.6 s
Last	0	3	0 s

packet and B2 relays. Although L2 moved out of the S2's transmission range it also can sends ACK to S2 by the way of B2. Accordingly, S2 could update its neighbor information. As described above, vehicles update topology information while broadcasting data and therefore the proposed protocol can efficiently handle mobility.

4.7 Consecutive Broadcasting: ACK and NACK

In many situations, a sender needs to broadcast many packets. We call it consecutive broadcasting. Actually, our proposed *MPR* selection algorithm tends to use the same relay nodes if other metrics of candidate nodes are equal. This makes consecutive broadcasting more possible. In case of consecutive broadcasting, we use negative ACK (NACK) to reduce the number of control messages. The sender sets the consecutive flag to 1 and set the next packet arrival time. Every sender should recalculate the next packet arrival time according to packet generation interval, contention delay and propagation delay. The sender also needs to attach a consecutive sequence number (seq no) to the packet. The consecutive sequence number is used to let downstream nodes know whether this is the first packet of consecutive broadcasting or not. If it is not the first packet, TO-ACK-LIST is not required. For example, in case of 3 consecutive broadcasting, the fields of consecutive broadcasting for first, second and last packet are shown in **Table 2**.

When a node receives a packet with consecutive flag equals one, the node

records the next packet arrival time and starts a timer. If the packet is the first packet of consecutive broadcasting, the node sends an ACK to the upstream node and otherwise not. If the next packet did not arrive before expected arrival time, the node sends an NACK to source node and source node would retransmit the packet upon reception of the NACK. If it is the last packet of consecutive broadcasting, the sender sets the consecutive flag to 0 in order to notify receivers not to wait for the next packet.

Generally, if the sender node (at IP layer) can predict the next packet generation time, consecutive broadcasting can be used. For example, applications that generate a constant rate stream can use consecutive broadcasting. Since the generation rate is constant, the sender node can predict the generation time of the next packet. Another case that can use consecutive broadcasting is when the application data size is larger than the maximum transmission unit. In that case application data will be divided to multiple IP datagrams and thus the sender node will be aware of the next packet scheduling time.

4.8 Boundary Selection Error Handling

A node may fail to select the boundary nodes. If nothing is selected, it might be because of following two reasons. One is because a rebroadcast cannot provide additional coverage. Another is because this node has insufficient two-hop neighbor information. In the first situation, the node rebroadcasts the packet with TTL equals one. It means every neighbor node can receive this packet, but will not rebroadcast. In the second situation, the node rebroadcast with NULL BOUNDARY-LIST. If a node receives a packet with NULL BOUNDARY-LIST, it rebroadcasts the packet.

5. Performance Evaluation

The proposed protocol reduces broadcast redundancy by means of a method in which only boundary nodes relay broadcast packets. Clearly, the protocol is effective in high-density networks. In sparse networks, our proposed protocol is resistant to channel loss because it incorporates a retransmission mechanism. In mobile scenarios, the proposed protocol can update topology information using ACKs without introducing too much overhead.

The proposed protocol uses a subset of neighbor nodes to forward a data packet.

Table 3 Sizes of fields in ACK message.

Field	Size
Destination node address	4 bytes
Broadcast source node address	4 bytes
Broadcast ID	4 bytes
ACK sender node address	4 bytes

In order to check the reception status of all receivers, we use explicit ACKs when they are required. The sizes of all fields in an ACK message can be seen in **Table 3**. In Table 3, Destination node address field is the address of the node this ACK should be sent to. ACK sender node address field is the address of the node that initiates the ACK. Upon reception of the ACK, a node can use ACK sender node address, Broadcast source node address and Broadcast ID to determine which node has received which packet.

In the proposed protocol, if a node will not forward the packet, it sends an explicit ACK to the sender node and otherwise not. That is, the number of explicit ACKs used in the proposed protocol is determined by how many nodes do not forward the packet. We can know that the number of packets used in the proposed protocol is same to that of flooding. Also according to IEEE 802.11 standard [13], broadcast frames shall not be fragmented even if their length exceeds the defined fragmentation threshold. Therefore, the number of MAC frames used in the proposed protocol is also the same to that used in flooding.

As far as the MAC frame size is concerned, the ACK frame size used in the proposed protocol is smaller than the data frame size of flooding. However, since the proposed protocol attaches additional information to the data packets, the MAC data frame size in the proposed protocol can be larger than in flooding. We show the sizes of additional information in **Table 4**. This raises a question of how the additional overhead affects the performance of the proposed protocol. We can use two facts to explain that this overhead is well compensated by the advantages of the proposed protocol. First, in flooding, many packets are dropped because collisions incurred from all neighbors try to rebroadcast a packet at the same time. The proposed protocol efficiently reduces the number of rebroadcasts, so collisions can be avoided. Second, because the ACK frame size is much smaller

Table 4 Sizes of additional information.

Field	Size
Source node address	4 bytes
Broadcast ID	4 bytes
BOUNDARY-LIST	4 bytes × list size
TO-ACK-LIST	4 bytes × list size
REVERSE-BOUNDARY-LIST	4 bytes × list size
Consecutive broadcasting flag	1 bit
Retransmit flag	1 bit
Retransmit source node address	4 bytes

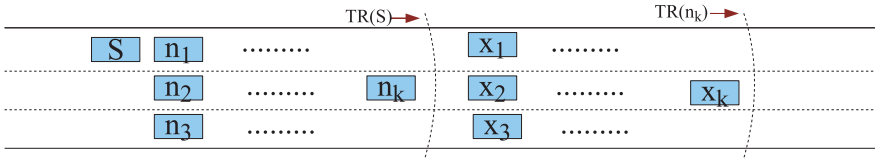

Fig. 7 An example for overhead analysis.

than the data frame size, we can easily know that the overall overhead of the proposed protocol will always be lower than flooding.

We here explain the effect of the additional overhead with an example, which can be seen in **Fig. 7**. In the figure, the source node S broadcasts a data packet and its one-hop neighbors relay the packet. For simplicity, we here only consider the overheads incurred by node S and its one-hop neighbors. However, the calculation given below can be easily extended to node S' two-hop neighbors and further.

In flooding, a data packet will be broadcasted by $k+1$ nodes (S, n_1, $\cdots$, n_k). In the proposed protocol, two nodes, S and n_k will broadcast the data packet and other nodes (n_1, $\cdots$, n_{k-1}) will send ACKs to node S. In both protocols, the number of total MAC frames will be $k+1$. However, the total frame sizes are different. In the flooding, when the application data size is 512 bytes, the MAC data frame size S_d will be $S_d = 512+20(\text{IPheader})+24(\text{MACheader})+4(\text{FCS}) = 560(\text{bytes})$. So, the total frame size will be $S_d \times (k+1)(\text{bytes})$ in flooding.

In the proposed protocol, MAC data frame size will be affected by the sizes of BOUNDARY-LIST, TO-ACK-LIST and REVERSE-BOUNDARY-LIST. We show the sizes of those lists in **Table 5**.

Table 5 Sizes of lists in data packets.

Sender	Field	Items	Size
S	BOUNDARY-LIST	$[n_k]$	4 bytes
	TO-ACK-LIST	$[n_1, \cdots, n_k]$	$4 \times k$ bytes
	REVERSE-BOUNDARY-LIST	[S]	4 bytes
n_k	BOUNDARY-LIST	$[x_k]$	4 bytes
	TO-ACK-LIST	$[x_1, \cdots, x_k]$	$4 \times k$ bytes
	REVERSE-BOUNDARY-LIST	[S, n_k]	8 bytes

As Table 5 shows, MAC frame size of the data packet sent by node S (S_{d1}) will be $S_{d1} = S_d + 13 + 4 + 4 \times k + 4 = 581 + 4 \times k(\text{bytes})$ where 13 is the total size of fixed length fields which includes source node address, Broadcast ID, Consecutive broadcasting flag, Retransmit flag and Retransmit Source node address. In above calculation, 4, $4 \times k$ and 4 are the sizes of BOUNDARY-LIST, TO-ACK-LIST and REVERSE-BOUNDARY-LIST respectively. MAC frame size of the data packet sent by node n_k (S_{d2}) will be $S_{d2} = S_d + 13 + 4 + 4 \times k + 8 = 585 + 4 \times k(\text{bytes})$. Similarly, ACK frame size (S_A) will be $S_A = 16 + 20(\text{IPheader}) + 24(\text{MACheader}) + 4(\text{FCS}) = 64(\text{bytes})$. So the total MAC frame size in the proposed protocol will be $S_{d1} + S_{d2} + S_A \times k = 581 + 4 \times k + 585 + 4 \times k + 64 \times (k-1) = 1102 + 72 \times k(\text{bytes})$. When k is 2, total MAC frame size of the flooding will be 1,680 bytes and total MAC frame size of the proposed protocol will be 1,246 bytes. When k is 32, total MAC frame size of the flooding will be 17,920 bytes and total MAC frame size of the proposed protocol will be 3,406 bytes. Therefore, we can see that total overhead of the proposed protocol is lower than flooding, especially in the high-density networks.

The above-given descriptions show that the proposed protocol always has lower overhead than flooding even in the extreme situation of one source only sending one packet. In the case where broadcast sources have more than one packet to broadcast consecutively (consecutive broadcasting), the proposed protocol benefits from a negative ACK mechanism. Since explicit ACK is only required in the reception of the first packet from the sender, the protocol's overhead decreases

notably.

In order to validate our analysis and further evaluate the proposed protocol's performance, we conducted simulations with ns-2 and show the simulation results. We assume every node has a transmission range of 250 m. We use omnidirectional antennas and TwoRayGround propagation model. IEEE 802.11 MAC [13] and 512 bytes sized data packets have been used. Other simulation parameters use default setting of ns2.28.

In order to capture the realistic character of vehicles' movements to our simulation, we use Mobility Generator described in Ref. 14). We use a freeway that has four lanes in two different directions. All lanes of the freeway are 2,000 m in length. Maximum velocity is 50 m/s and every vehicle accelerates at the rate of ten percent of the maximum allowable velocity if there are no other vehicles ahead of it. In our protocol, the broadcast source node uses algorithm 2 and other sender nodes use algorithm 1 to relay data packets. The sizes of ACK and additional information used in the proposed protocol can be seen in Table 3 and Table 4. All data presented in this paper are the average value of simulations repeated 10 times with different node movements.

5.1 Performance of the Enhanced *MPR* Selection Algorithm

We first evaluate the effectiveness of proposed *MPR* selection algorithm. It is possible that selected *MPRs* fail to receive data because of vehicles' movements. **Figure 8** shows the success ratio of original *MPR* selection algorithm [8] and the proposed enhanced *MPR* selection algorithm for various maximum velocities. Because two-hop neighbor information is updated on reception of hello messages, we use two different hello intervals of 0.5 s and 1 s. We use 200 nodes to acquire enough mobility. In order to evaluate the effect of the enhanced *MPR* selection more correctly, we do not use retransmission mechanisms in this simulation.

From simulation results, we observe that original *MPR* selection algorithm's success ratio decreases drastically with the increasing of node velocity especially in 1 s hello interval. This is because the original *MPR* selection algorithm may select the nodes moving toward different directions as *MPR* nodes. Those nodes always fail to relay packets successfully because of the vehicles' movements. However, as a result of including mobility prediction in the *MPR* selection procedure, the enhanced *MPR* selects relatively stable nodes and therefore can achieve high

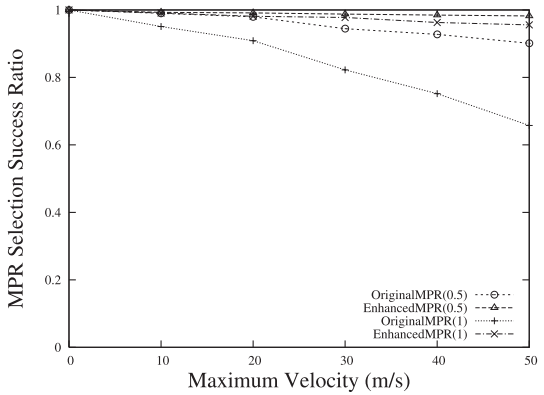

Fig. 8 Success ratio for various maximum velocities.

success ratio regardless of node velocity and hello interval. Simulation results confirm that it is important to consider vehicles' mobility in *MPR* selection.

5.2 Effect of Node Density

In order to evaluate the effect of node density with the proposed protocol, we use various number of nodes ranging from 100 to 500. The proposed protocol is compared with flooding and other three VANET broadcast protocols (weighted p-persistence, slotted 1-persistence and slotted p-persistence with four slots) proposed in Ref. 6). We use weighted p-persistence, slotted 1-persistence and slotted p-persistence scheme because they are efficient and recent broadcast suppression techniques in VANETs.

As for the flooding, it can be seen in **Fig. 9** that delivery ratio decreases drastically with increasing node density. This is due to the broadcast storm problem of flooding. There is a high probability that many nodes that very close to the sender node will try to rebroadcast. Therefore, many collisions occur because of the lack of RTS/CTS.

The weighted p-persistence, slotted 1-persistence and slotted p-persistence achieve better performance than flooding in a high-density network due to the reduction of rebroadcasts. We show the number of rebroadcasts of the protocols in **Fig. 10**. We can observe that our proposed protocol is more efficient than other protocols. In the proposed protocol, the sender node specifies the bound-

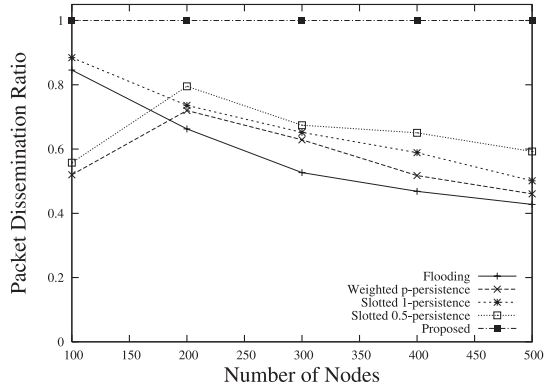

Fig. 9 Packet delivery ratio for various node density.

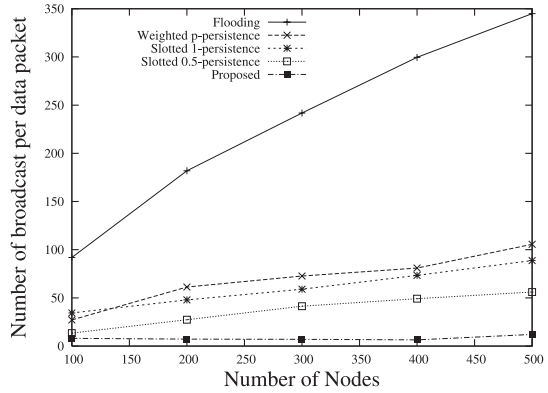

Fig. 10 Number of broadcast for various node density.

ary nodes and only boundary nodes rebroadcast. The proposed protocol benefits from unicast ACKs and retransmission mechanism and therefore can acquire one hundred percent delivery ratio.

For protocol overhead, at the worst case of no consecutive broadcasting, the proposed protocol's total packet number is near to that of flooding. However, ACKs are smaller than data packets. Hence, total overhead of the proposed protocol is lower than flooding. We show MAC overhead comparison of the protocols

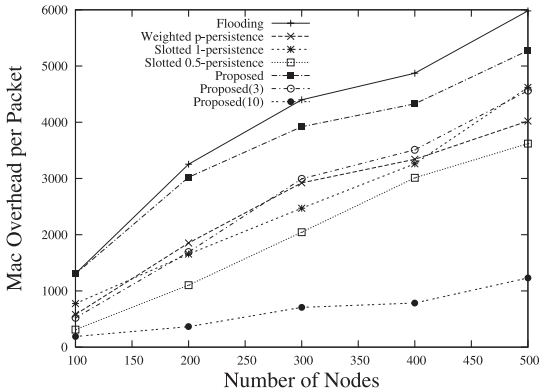

Fig. 11 MAC overhead per packet for various node density.

in **Fig. 11**. In Fig. 11, we use line *Proposed* to mean the proposed protocol with no consecutive broadcasting. Line *Proposed*(3) denotes senders utilizing consecutive broadcasting for every 3 packets and line *Proposed*(10) denotes senders utilizing consecutive broadcasting for every 10 packets. In this paper, MAC overhead is simply calculated as the number of sent or received MAC layer frames.

We can observe from Fig. 11 that the proposed method performs lower MAC overhead than flooding. The saved rebroadcasts (Fig. 10) in the proposed protocol can explain this effect. In the case of no consecutive broadcasting, the proposed protocol has higher MAC overhead than the weighted p-persistence, slotted 1-persistence and slotted p-persistence scheme due to ACK messages and retransmission mechanism. In case of consecutive broadcasting, receivers only need to explicitly acknowledge the first packet. Therefore, the proposed protocol shows notably lower overhead. It is shown that although proposed method includes ACK messages to improve transmission reliability, this does not significantly increase overhead because those messages are sent unicast. In short, the proposed protocol can significantly improve reliability while keeping MAC overhead at an acceptable level.

5.3 Performance over Sparse Networks

A novel VANET broadcast protocol also should work well in sparse networks. Flooding may be considered as an acceptable broadcast scheme in sparse net-

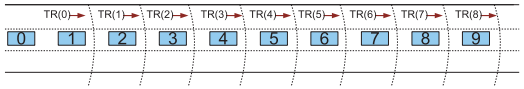

Fig. 12 Topology of the sparse Network.

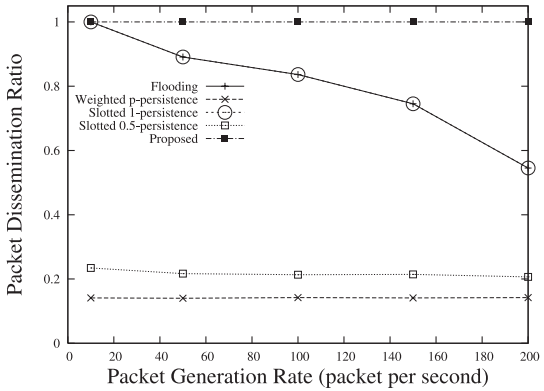

Fig. 13 Packet delivery ratio for various packet generation rate.

works. Interestingly, we observe that if the packet transmission rate is high enough, simple flooding will be confronted with large number of collisions even in the sparse networks. We generated a sparsely connected single lane network (**Fig. 12**) to simulate this effect. There are 10 nodes distributed in a chain manner. Besides the first node and the last node, every node has two neighbors. The first node and the last node have only one neighbor. The distance between two neighbor nodes is 200 m. Because the transmission range is 250 m, this network is fully connected. Simulation results are plotted in **Fig. 13**.

We can see that flooding performs poorly when the packet generation rate is high. In the slotted 1-persistence scheme, a node rebroadcasts with probability 1 at the assigned time slot. Hence, the slotted 1-persistence scheme works similar to the flooding in sparse networks. In this simulation, every two neighboring vehicles' distance is near the transmission range. A node rebroadcasts the packet immediately after the reception of the data packet. Hence, the slotted 1-persistence scheme works exactly same to the flooding. As for the weighted p-

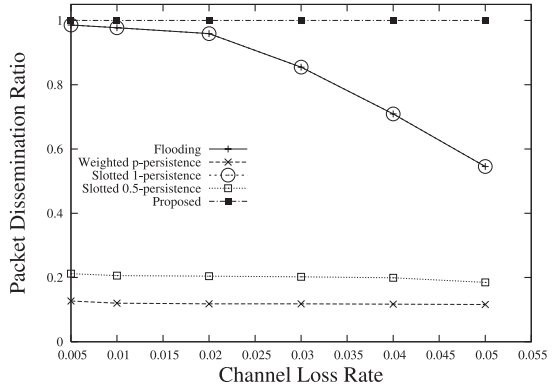

Fig. 14 Packet delivery ratio for various channel loss rate.

persistence and slotted 1-persistence, they behave poor performance in a sparse network because of probabilistic broadcasting. However, thanks to retransmission mechanism, the proposed protocol can achieve one hundred percent delivery ratio.

We also simulate the protocols' performance over different channel loss rate. We use low data rate of ten packets per second. If there is no loss in the wireless channel, the flooding can achieve perfect delivery ratio. However, in the loss channel, as shown in **Fig. 14**, we know that the flooding cannot achieve enough penetration. It is obvious that retransmission is required in sparse networks and the proposed protocol benefits from doing so.

5.4 Delay

Dissemination delay is an important metric to evaluate a broadcast protocol's performance. The messages should be delivered to intended receivers within the given time. However, flooding cannot disseminate messages quickly enough because of too many redundant rebroadcasts. We use 400 nodes in this simulation and show the delay comparison of the protocols in **Fig. 15**.

We find that the proposed protocol achieves lowest delay because of the following reasons. The proposed protocol uses boundary nodes to rebroadcast the packets. Consequently, the proposed protocol reduces the number of hops to the desired receivers. The proposed protocol also reduces the number of rebroadcasts

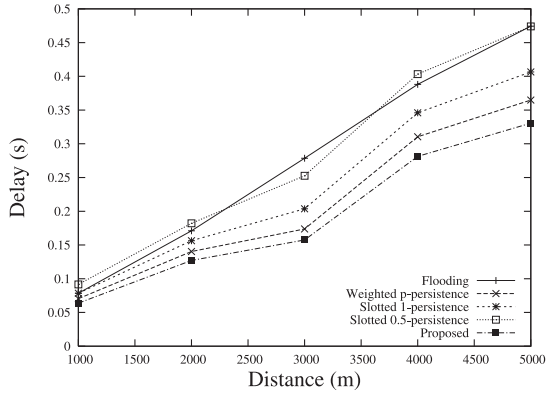

Fig. 15 Delay for various distances.

and therefore results decreasing contention time.

In flooding, however, the nodes that provide larger progress on distance possibly lose the data packets due to packet collisions. As a result, the packets are delayed because they are delivered through sub-optimal paths (longer paths). For the weighted p-persistence, slotted 1-persistence and slotted p-persistence schemes, we can observe that they perform with acceptable delays. But, they significantly suffer from long delays in a sparse network, due to the scheduling and waiting time required before rebroadcast [6].

6. Conclusions

Reliability is the most important issue in vehicular safety message dissemination. In this paper, we proposed a multi-hop broadcast protocol, which can ensure strict reliability. We use an efficient acknowledgement method to detect whether all desired receivers have received the packet. To mitigate broadcast storms, the proposed protocol uses boundary nodes to relay data packets. In boundary node selection, we use an enhanced *MPR* algorithm, which is also proposed in this paper.

We used simulations to further evaluate the protocol's performance. Simulation results confirmed that the proposed protocol has notable performance improvement in various traffic conditions compared to other broadcast methods. In summary, the proposed protocol provides an efficient reliable broadcast solution to disseminate safety messages in vehicular ad hoc networks.

References

1) The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/
2) Tonguz, O., Wisitpongphan, N., Bai, F., Mudalige, P. and Sadekar, V.: Broadcasting in VANET, *Proc. Mobile Networking for Vehicular Environments*, pp.7–12 (2007).
3) Liu, Y., Li, F.Y. and Schwefel, H.-P.: Reliable Broadcast in Error-Prone Multi-hop Wireless Networks: Algorithms and Evaluation, *Proc. IEEE Global Telecommunications Conf.*, Washington, USA, pp.5329–5334 (2007).
4) Jiang, H., Guo, H. and Chen L.: Reliable and Efficient Alarm Message Routing in VANET, *Proc. 28th Intl. Conf. Distributed Computing Systems Workshops*, Beijing, China, pp.186–191 (2008).
5) Khakbaz, S. and Fathy, M.: A Reliable Method for Disseminating Safety Information in Vehicular Ad Hoc Networks Considering Fragmentation Problem, *Proc. Fourth Intl. Conf. Wireless and Mobile Communications*, Athens, Greece, pp.25–30 (2008).
6) Wisitpongphan, N. and Tonguz, K.O.: Broadcast Storm Mitigation Techniques in Vehicular Ad Hoc Networks, *IEEE Wireless Communications*, Vol.14, No.6, pp.84–94 (2007).
7) Blaszczyszyn, B., Laouiti, A., Muhlethaler, P. and Toor, Y.: Opportunistic Broadcast in VANETs (OB-VAN) Using Active Signaling for Relays Selection, *Proc. 8th Intl. Conf. ITS Telecommunications*, Phuket, Thailand, pp.384–389 (2008).
8) Qayyum, A., Viennot, L. and Laouiti, A.: Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks, *Proc. 35th Annual Hawaii Intl. Conf. System Sciences*, Big Island, Hawaii, pp.3866–3875 (2002).
9) Wu, J. and Dai, F.: A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks, *IEEE Trans. Comput.*, Vol.53, No.10, pp.1343–1354 (2004).
10) Khabbazian, M. and Bhargava, V.K.: Efficient Broadcasting in Mobile Ad Hoc Networks, *IEEE Trans. Mobile Computing*, Vol.8, No.2, pp.231–245 (2009).
11) Wu, J., Lou, W. and Dai, F.: Extended Multipoint Relays to Determine Connected Dominating Sets in MANETs, *IEEE Trans. Comput.*, Vol.55, No.3, pp.334–347 (2006).
12) Clausen, T. and Jacquet, P.: Optimized Link State Routing Protocol (OLSR), RFC 3626 (Oct. 2003).
13) IEEE Std 802.11.: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, *ANSI/IEEE 802.11 Std* (Aug. 1999).
14) Bai, F., Sadagopan, N. and Helmy, A.: Important: A Framework to Systematically Analyze The Impact of Mobility on Performance of Routing Protocols for Adhoc

Networks, *Proc. 22nd Annual Joint Conf. IEEE Computer and Communications Societies*, San Francisco, USA, pp.825–835 (2003).

(Received May 25, 2009)
(Accepted December 17, 2009)
(Released March 10, 2010)

Celimuge Wu received his M.E. degree from Beijing Institute of Technology, Beijing, China, in 2006. He is currently a Ph.D. candidate at Department of Information Network Science, Graduate School of Information Systems, University of Electro-Communications, Tokyo, Japan. His current research interests include mobile ad hoc networks, networking architectures and protocols. He is a member of IPSJ and IEICE.

Kazuya Kumekawa received his B.S. degree in engineering, M.S., and Ph.D. degrees in science from Tohoku University in Japan, in 1992, 1994, and 1997, respectively. He is currently an assistant professor at Department of Information Network Science, Graduate School of Information Systems, University of Electro-Communications in Tokyo, Japan.

Toshihiko Kato received his B.E., M.E. and Dr.Eng. degrees electrical engineering from the University of Tokyo, in 1978, 1980 and 1983, respectively. He joined KDD in 1983 and worked in the field of communication protocols of OSI and Internet until 2002. From 1987 to 1988, he was a visiting scientist at Carnegie Mellon University. He is now a professor of Graduate School of Information Systems in University of Electro-Communications in Tokyo, Japan. His current research interests include protocol for mobile Internet, high speed Internet and ad hoc network.