A Water-Wave Broadcast Scheme for Emergency Messages in VANET

Shou-Chih Lo[1*] Jhih-Siao Gao[1] Chih-Cheng Tseng[2]

[1]*Department of Computer Science and Information Engineering*
National Dong Hwa University
Hualien, Taiwan, ROC
[2]*Department of Electrical Engineering,*
National Ilan University
I-Lan, Taiwan, ROC

sclo@mail.ndhu.edu.tw
Tel: +886-3-8634029
Fax: +886-3-8634010

Abstract—Vehicular Ad Hoc Network (VANET) is an emerging new technology and a promising platform for the Intelligent Transportation System (ITS). The most important application of VANET is disseminating emergency messages to drivers in case of dangerous events. The effectiveness depends on the design of a broadcast scheme. A simple broadcast scheme encounters many problems such as broadcast storm, connection hole, building shadow, and intersection problems. In this paper, we propose an efficient broadcast scheme that simulates water wave propagation to spread emergency messages. This scheme provides warning services with both space and time constraints. Most existing broadcast schemes provide inadequate strategies for limiting the time period of a warning. We verified the performance of our proposed scheme in a simulated street environment with vehicle movements to show the superiority of this scheme in high broadcast coverage areas.

Keywords-VANET; ITS; Emergency Broadcast; Carry and Forward

1. Introduction

Intelligent Transportation Systems (ITSs) integrating advanced computer, communication, and sensing technologies provide efficient traffic management and real time traffic status for the safety and comfort of drivers. ITS can greatly reduce vehicle accidents, traffic congestion, and personnel cost in traffic management. With recent advances in wireless technology, the introduction of wireless communication into ITS has allowed development of the Vehicular Ad Hoc Network (VANET) [1].

---

[*] To whom all correspondence should be sent.

VANET has received considerable attention in recent years, and the related standards and applications are promoted in many countries. For example, the US Federal Communications Commission (FCC) in 1999 decided to allocate the frequency band 5.85GHz~5.925GHz for Dedicated Short Range Communication (DSRC). In 2004, the IEEE initiated task group 802.11p to study an amendment to the IEEE 802.11 standard, adding Wireless Access in the Vehicular Environment (WAVE). The physical layer is based on IEEE 802.11a and the MAC layer is modified from IEEE 802.11e. IEEE 1609 is a family of high layer standards which rely on IEEE 802.11p. These standards include resource management, security services, network services, and multichannel operations. In Europe, six major automobile companies that form the Car2Car Communication Consortium (C2CCC) [2] initiated the NOW project [3] in 2004. Other projects such as FleetNet [4] and VICS [5] are run in Germany and Japan, respectively.

VANET is a specialized Mobile Ad Hoc Network (MANET) that connects vehicles and roadside facilities. VANET provides both Roadside-to-Vehicle Communication (RVC) and Inter-Vehicle Communication (IVC). RVC supports Internet access and data retrieval services for vehicles and IVC relays data among vehicles. The most important application of VANET [6] is to disseminate emergency messages to drivers in case of dangerous events such as slippery roads or car accidents. A warning message needs to be delivered with low delay and high reliability to those vehicles that are located within a specified warning area.

Typically, data dissemination in wireless networks is achieved in a flood using successive broadcasts. However, wireless Medium Access Control (MAC) protocols such as IEEE 802.11 are not efficient enough, because signal interference, collision, and contention create numerous problems with data transmission such as hidden terminal [7] and broadcast storm problems [8]. In a vehicular environment, the highly dynamic feature of vehicle movement and the signal shadow effect from surrounding buildings make data dissemination even more challenging.

In this paper, we first discuss how the requirements for emergency broadcast in VANET differ from that for data broadcast in MANET. We then detail the weaknesses of conventional broadcast schemes in vehicular environments by indicating three new challenges: connection hole, intersection, and shadow problems. A comprehensive survey of VANET broadcast is provided, including discussion of design phases and design philosophies. A comparison table of several existing emergency broadcast schemes is given as well. We realize that these existing schemes cannot fully and efficiently solve the new challenges. The most important consideration is that these schemes cannot satisfy the typical requirements of an emergency warning service given both space and time constraints. Most schemes provide inadequate strategies for constraining a warning event to a certain area and time period. This survey result is an extension of our previous conference paper [9], but includes a more complete survey and proposal of a new broadcast scheme.

Our proposed scheme simulates water wave propagation that spreads emergency messages along the way while avoiding the aforementioned problems. The concepts of head waves, aftereffect waves, and ripple effects are introduced to explain our design. The processing steps are clearly divided into different parts by identifying different node roles. These make our proposed scheme easy to understand and implement. We verified the performance of our scheme in a simulated street environment with vehicle movements to show the superiority of this scheme in high broadcast coverage areas.

The remainder of this paper is organized as follows. A brief survey of different broadcast schemes is given in Section 2, followed by our proposed scheme in Section 3. We evaluate the performance of our proposal in Section 4, and offer a brief conclusion in Section 5.

2. Related Work

We will first discuss the shortcomings of traditional wireless broadcast approaches in VANET and then classify existing approaches by identifying their different design concepts.

2.1 Difference from traditional problems

Broadcasting is an important function in mobile wireless networks, including routing, service discovery, and data dissemination. Simple flooding, where each node rebroadcasts each received broadcast packet, has a scalability problem and may suffer from a serious broadcast storm problem. In the broadcast storm, several nodes rebroadcast the same packet at the same time, causing highly redundant transmission, channel contention and collision. To overcome this problem, several broadcast schemes have been proposed particularly for MANET environments.

In [10], these schemes are classified into: probability based, area based, and neighbor knowledge based methods. With the probability based method, whether a particular node rebroadcasts a packet is based on a certain probability. With the area based method, the farthest one-hop neighbor of the current broadcast node is commonly selected as a rebroadcast node, since this node provides the largest additional broadcast coverage area. With the neighbor knowledge based method, the list of one-hop and even two-hop neighbors is exchanged among neighboring nodes using “hello” packets. Whether a node rebroadcasts is decided by checking whether all its one-hop neighbors have already received the designated broadcast packet from other neighboring nodes. The concept of neighbor elimination is proposed in [11], by which a node need not rebroadcast a packet if all its one-hop neighbors have been covered in the transmission ranges of previous broadcasts.

A more advanced concept based on neighbor knowledge has been studied by constructing a so-called connected dominating set [11]. All nodes in the network are either nodes in the set or one-hop neighbors of nodes in the set. Only the nodes in the dominating set have the responsibility to rebroadcast. However, the construction of an optimal dominating set is a NP-complete problem. Some heuristics have been proposed by globally or locally exchanging neighbor knowledge information such as the number or locations of neighbors. A complete survey of broadcast schemes in MANET is given in [12].

Although emergency warning services in VANET can be achieved using the same broadcasting technique, traditional solutions are not sufficient. The key point is that traditional solutions provide network-wide message dissemination to all connected nodes in the network. However, emergency warning needs solutions that provide area-wide message dissemination to all nodes within a certain area even though they are not connected. Unfortunately, network partition or topology fragmentation frequently occurs in vehicular environments due to sparse vehicle distribution and signal shadowing from roadside buildings. Therefore, traditional solutions become unsuitable.

In the example shown in Fig. 1, car A broadcasts an emergency message, but car B cannot receive this message because it is out of the current communication range of car A. That is, the network topology is fragmented. This is called a *connection hole problem*. A similar case happens for car C due to the signal shadowing from buildings at the corner. This is called *building shadow problem*. If car D is the only rebroadcast node for this emergency message, car E, which can hear the broadcast from car A, will not further forward the emergency message to its neighbors. As a result, car F misses this message. This situation usually occurs in an intersection area, so this is called an *intersection problem*. Consequently, the challenges of broadcast in VANET include the connection hole, building shadow, and intersection problems in addition to the broadcast storm problem.

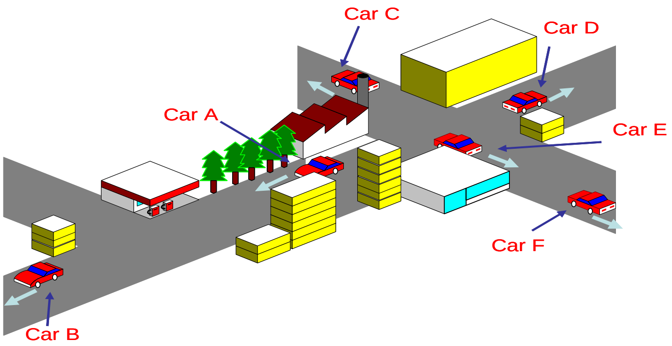


Fig. 1. Broadcast problems in an urban environment.

Beyond the difference between network-wide achievement and area-wide achievement, the design focus in VANET has other differences from MANET. Fast dissemination and high reliability are two major concerns in VANET. Fast dissemination is essential so that drivers can be notified of an event quickly so as to have plenty of time to react. High reliability allows as many relevant drivers as possible to be notified of an event. However, lowering dissemination costs to reduce power use of devices is the most concern in MANET.

2.2 Comparison of existing solutions

Before introducing and comparing existing broadcast schemes for VANET, we will first identify the general phases in broadcast design. In each phase, we discuss different methods with their strengths and weaknesses. After receiving an emergency broadcast message, a node typically performs the following procedures in sequence:

(1) **Message confirmation**: Each node needs to confirm whether the message is a redundant one and whether the message is an irrelevant one. To achieve this, certain data are recorded at each node and in the message header.

(2) **Waiting delay**: To prevent redundant rebroadcast, one node needs to wait for a certain delay time during which it can listen to the activities of other neighboring nodes so it can make a more accurate decision. Thus, a suitable waiting delay function is necessary.

(3) **Rebroadcast decision**: This is the core of any broadcast scheme, that determines whether one node needs to rebroadcast a message or not. The number and positions of rebroadcast nodes are important issues for reducing the effects of broadcast storm, shadowing, and intersection problems. A criterion-based or probability-based method is usually applied.

(4) **Reliability checking**: To recover from any failure of rebroadcast that would stop further message dissemination, an action that detects and compensates for errors is necessary.

(5) **Connection hole prevention**: A method that can detect and overcome connection holes is applied here. Data buffering is a typical technique.

Any broadcast scheme that involves a waiting delay will rebroadcast after a variable delay. In this kind of scheme, an early rebroadcast can cancel a later one that is unnecessary. On the other hand, a later rebroadcast can compensate for the failure of an early one. These features allow for the advantages of low redundancy and high reliability.

In the rebroadcast decision phase, the decision is made either at the receiver side (*receiver-initiated decision*) or at the sender side (*sender-initiated decision*). One node decides by itself whether to be a rebroadcast node or not if the former method is used. However, the current broadcast node decides which of its neighbors should become rebroadcast nodes if the latter method is used. The decision is embedded in the emergency message, which causes overhead in transmission. The number of rebroadcast nodes is an issue of concern at each decision. Fewer rebroadcast nodes can decrease the broadcast storm problem but will increase the intersection problem and reduce the transmission reliability.

To reduce the effects of intersection and shadowing problems, three methods can be applied. One way (*close intersection approach*) is selecting a rebroadcast node that is closer to the intersection, since this node has a better signal coverage of the other road segments. The weakness of this approach is that the locations of intersections must be known. Another way (*roadside node approach*) relies on roadside units that are installed in intersections. These roadside units act as relay nodes that enable mobile nodes passing by them to upload and download data. This approach involves a higher hardware installation cost. The third approach (*node diversity approach*) is selecting diverse nodes moving in different directions, since these nodes may drive into different road segments when passing an intersection.

The decision criterion can be based on either *sender/receiver knowledge* or *neighbor knowledge*. Sender/receiver knowledge is the information about a pair of sender and receiver nodes. One important criterion is the relative distance between them. Neighbor knowledge is the information exchanged by neighboring nodes by exchanging hello packets. Neighbor elimination as introduced before is based on such neighbor knowledge. Certainly, more information supports a more accurate decision but costs more to exchange.

Reliability checking is necessary for environments with communication errors. A typical solution relies on the current broadcast node to broadcast the previous message again once a broadcast error is detected. The error detection can be performed in an implicit or explicit way. An implicit way is by listening to any rebroadcast of the same message from neighboring nodes after a certain time period. An explicit way is to force a rebroadcast node to reply an ACK packet to the current broadcast node.

The connection hole prevention phase is particularly necessary for partially connected networks. To propagate messages among network fragments, the *carry and forward* mechanism [13] is commonly used. A mobile node can carry messages received from one node in one network fragment and forward them to one node in each of the other network fragments. This mechanism needs the support of buffer space of each node.

Table 1. Comparisons of various emergency broadcast schemes.

Schemes	Waiting Delay	Rebroadcast Decision	Shadowing and Intersection Solutions	Persistence Policy	Reliability Checking	Connection Hole Prevention
WPP [14]	Fixed	Sender/Receiver Knowledge, Receiver-initiated	No	Weighted *p* persistence	Timeout/Implicit	No
SPP [14]	Sector-based	Sender/Receiver Knowledge, Receiver-initiated	No	*p* persistence	Timeout/Implicit	No
S1P [14]	Sector-based	Sender/Receiver Knowledge, Receiver-initiated	No	1 persistence	Timeout//Implicit	No
ODAM [15]	Distance-based	Sender/Receiver Knowledge, Receiver-initiated	No	1 persistence	No	Carry and Forward
EDB [16]	Distance-based	Sender/Receiver Knowledge, Receiver-initiated	Road Side Units	1 persistence	Timeout/Explicit	Carry and Forward
DP-IB [17]	Distance-based	Neighbor Knowledge, Sender-initiated	Road Side Units	1 persistence	Timeout/Explicit	No
REAR [18]	Reliability-based	Neighbor Knowledge, Receiver-initiated	No	1 persistence	No	No
PCC [19]	No (Primary node)/Fixed (Candidate node)	Neighbor Knowledge, Sender-initiated	No	1 persistence	Timeout/Explicit	No
RBM [20]	Distance-based	Sender/Receiver Knowledge, Receiver-initiated	Node Diversity	1 persistence	Timeout//Implicit	Carry and Forward
V-TRADE [21]	No	Neighbor Knowledge, Sender-initiated	No	1 persistence	No	No
VRR [22]	Motion-based	Neighbor Knowledge, Sender-initiated	No	1 persistence	No	No
SB [23]	Sector-based	Sender/Receiver Knowledge, Receiver-initiated	No	1 persistence	Timeout/Explicit	No
UMB [24]	Sector-based (Jamming)	Sender/Receiver Knowledge, Receiver-initiated	Road-Side Node	1 persistence	Timeout/Explicit	Carry and Forward
AMB[24]	Sector-based (Jamming)	Sender/Receiver Knowledge, Receiver-initiated	Close Intersection	1 persistence	Timeout/Explicit	Carry and Forward
WWB	Distance-based	Neighbor Knowledge, Receiver-initiated	Node Diversity	*p* persistence	Timeout//Implicit	Carry and Forward

The features of some existing broadcast schemes of VANET are summarized in Table 1. The length of waiting time can be assigned by the fixed, distance-based, motion-based, sector-based, or reliability-based method. With the distance-based method, the length of the waiting time of a rebroadcast node is inversely proportional to the distance between this node and the current broadcast node. That is, a furthest node from the current broadcast node enters the rebroadcast decision phase earlier. With the motion-based method, a more complex function involving distance, moving speed, and moving direction is used. With the sector-based method, the distance values to the current broadcast node are classified into different sectors or ranges. The waiting time of a node is dependent on which sector this node is located in. A sector can be a road segment of fixed length in street space, or a donut-like circle of fixed width in open space. All nodes in the same sector wait for the same time period. Moreover, a node in a distant sector waits for a shorter time period than a node in a near sector. With the reliability-based method, the sum of receipt probabilities of neighboring nodes is used. Basically, a node with more neighbors has higher precedence as a rebroadcast node.

Most waiting delays are performed at layer 3 in the network protocol stack. However, the waiting is particularly performed at layer 2 in VRR, SB, UMB, and AMB. VRR and SB control the backoff timer. UMB and AMB send jamming signals of different lengths to distinguish different priorities. The weakness of these approaches is that the medium access control protocol has to be modified.

When a node has the chance to be a rebroadcast node, it follows one of the following persistence policies: p-persistence, weighted p-persistence, or 1-persistence. In other words, the probability of being a rebroadcast node is either p or 1. The p value is either assigned a fixed value or computed by a function. In WPP, this value is directly proportional to the distance between this node and the current broadcast node.

Most broadcast schemes select the farthest neighbor of the current broadcast node as the single rebroadcast node. However, the farthest two neighbors are selected in PCC as primary and candidate nodes which work in a cooperative way to increase reliability. V-TRADE also selects two rebroadcast nodes that are located ahead and behind the current broadcast node. ODAM uses an extreme method, which simply lets all nodes periodically broadcast the received emergency messages, to solve the connection hole problem.

3. Broadcast Scheme

In this paper, we propose a *Water-Wave Broadcast* (WWB) scheme that simulates the propagation of emergency messages using the analogy of water waves. The characteristics used in WWB are also shown at the last entry of Table 1. As shown in Fig. 2, the water wave is triggered at the center, and the head wave leads the wave propagation to the outer area. The aftereffect waves make the inner area rippling.

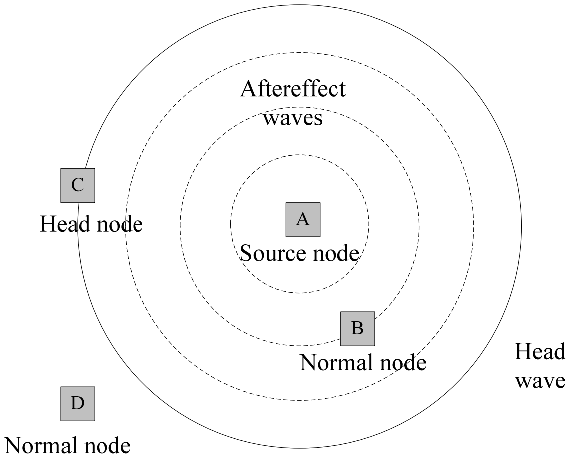


Fig. 2. Imaginary wave propagation.

In an emergency warning service, a warning wave is generated as an emergency event is detected. The propagation medium is vehicles on the road. The head wave spreads the event to the whole warning area. Vehicles hold the head wave as they move and forward it to other encounter vehicles. The warning area keeps rippling until the end of the warning time. Any node entering into a ripple area will be notified of this event. The following types of nodes (or node roles) are defined and some of them are depicted in Fig. 2:

- *Source node*: a node that triggers an emergency event.
- *Head node:* a node that promotes the head wave.
- *Normal node*: a general node in the network.
- *Sender node*: a node that is sending out a packet.
- *Receiver node*: a node that is receiving a packet.

First, we introduce the data structures used in our proposed scheme as listed below:

(1) **Emergency Broadcast Packet**: Source ID | Sequence Number | Warning Area | Warning Time | Event Content | Sender Location | TTL

(2) **Basic Hello Packet**: Node ID | Sequence Number | Node Location

(3) **Extended Hello Packet**: Node ID | Sequence Number | Node Location | Event List

(4) **Neighbor Table**: Node ID | Recent Hello Time | Location History | Estimated Motion

(5) **Event Table**: Source ID | Sequence Number | Warning Area | Warning Time | Event Content | Sender Location

An emergency event is enclosed in an emergency broadcast packet. The source ID field of this packet records the identification of the source node of this event. The combination of the source ID and sequence number fields uniquely identifies an emergency broadcast packet. The warning area field indicates the coordinates of a region where drivers should keep on the alert. The warning time filed indicates the point of time when the emergency event is expected to be clear. The event content field provides information about the event (type, degree, etc.). The sender location field denotes the location of the node that is currently sending out this broadcast packet. The TTL field indicates how far in hop count the broadcast packet will be propagated. One-hop local broadcast is obtained by setting TTL to be 1. An emergency event is *valid* for a node if the warning time is not expired and this node is currently located within the warning area.

Each node periodically announces its status to all its one-hop neighbors by broadcasting a hello packet. The cycle time is called a *hello interval*. Hello packets have two formats: basic and extended. These packets in both formats carry the current location of a node which is acquired from the GPS (Global Positioning System). An extended hello packet additionally carries an event list that summaries what emergency events have been received and are still valid for a node. This extended hello packet is used only when a node finds new neighbors. The periodic broadcasting of hello packets is not an extra overhead but a default operation provided in the IEEE 802.11p protocol for service announcements and car collision avoidance.

Moreover, each node maintains two tables: neighbor table and event table. The neighbor table records the status of its one-hop neighbors by listening to basic/extended hello packets from them. The recent hello time field records the time when a recent hello packet from a neighbor is received. If the elapsed time has a duration twice as long as one hello interval, this neighbor is supposed to be moving away and is removed from the neighbor table. The successive location data received from a neighbor are recorded in the field of location history by which we can estimate the motion vector (i.e., moving speed and direction) of this neighbor by exponential moving average.

The event table records valid emergency events that have been received or generated. Each entry has almost the same data fields as the emergency broadcast packet except for the TTL field. An event list is generated by listing the pair of source ID and sequence number fields of each entry in the event table.

Next, we will introduce how the WWB scheme works by explaining the mission of each node and the transition between different node roles. Basically, every node in the network always retains its role as a normal node. A normal node that experiences an aftereffect event wave would retain the event and make a ripple. An external event causes a normal node to bifurcate, initiating a new process acting as another role. After the mission of this new role is finished, the lifetime of the new role is over. The role transition diagram is shown in Fig. 3.

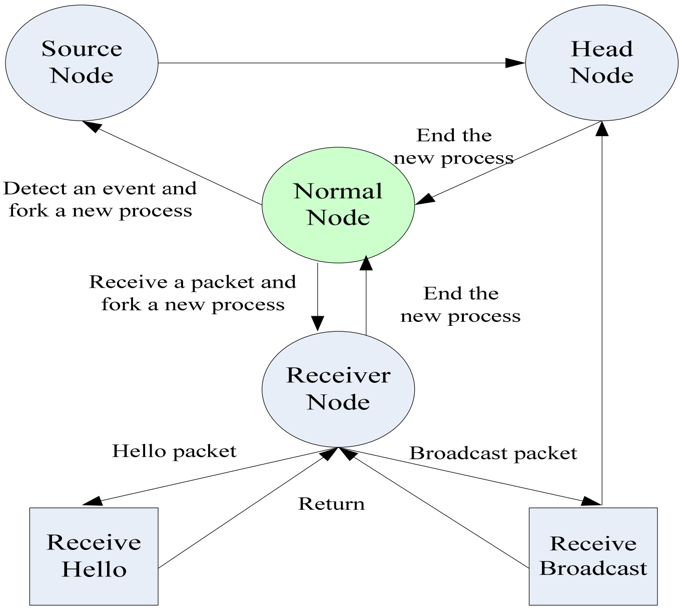


Fig. 3. Role transition diagram.

The mission of a normal node (see procedure NormalNode, below) is to periodically send hello packets and refresh the event and neighbor tables by removing obsolete data. Note that the hello packet in extended format is sent once as each new neighbor is found. A new neighbor is found when a node receives a hello packet from a node not existing in the current neighbor table. If an emergency event is detected by a normal node, the new role of being a source node is created. If any packet is received by a normal node, the new role of being a *receiver node* is created.

The mission of a source node (see procedure SourceNode, below) is to generate a new emergency broadcast packet and then change the node role into a head node. The mission of a head node (see procedure HeadNode, below) is to carry an emergency broadcast packet and forward it by broadcasting to neighbors. The carry and forward mechanism helps to solve the connection hole problem. After the broadcast, the head node performs a *reliability check* to confirm that any neighbors continue rebroadcasting the packet. One way of doing the reliability check is to rebroadcast the packet once again if without listening to any rebroadcast of the same packet from neighbors after a time period. This time period should have length at least equal to twice the average one-hop communication delay. Next, the head node decides whether to retain the role or not. In a certain situation as shown in Fig. 4, we would end the role of a head node if two same event propagations are encountered. A head node detects this situation by finding two (a pre-configured parameter) successive new encounter neighbors already having received this event. Another case for a normal node to be a head node is when this node receives an emergency message and decides to further forward the message.

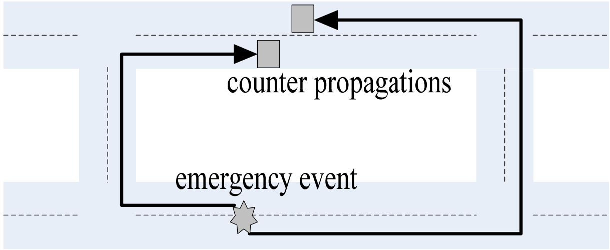


Fig. 4. Two cancellation propagations.

Next we will discuss the condition for a node to be a head node, beginning with the following definition.

Definition: *A head node is a node that is located on the border of current event propagation and cannot temporally spread the event forward or backward. A head node H needs to satisfy the condition: There is no neighbor of H that is ahead of or behind H and driving a similar direction as H.*

In Fig. 5, node S is broadcasting an emergency message and nodes A, B, C, D are receiver nodes. In this case, nodes A, B, and C can become head nodes, but node D can not due to its neighbor E. Nodes A and B can further spread the event along their routes without an intersection problem. When another group of nodes catches up with node C, node C would spread the event backward. Node D just informs node E with the event.

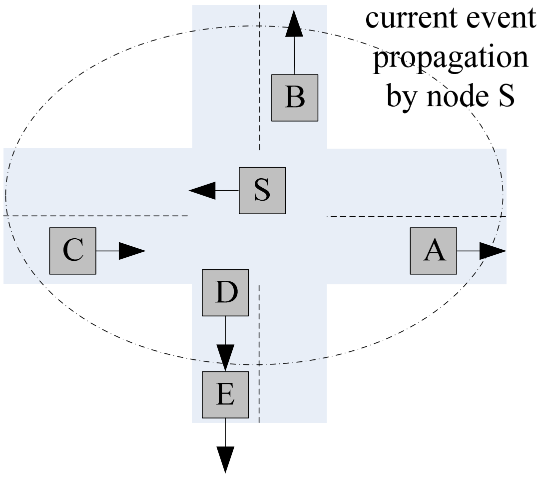


Fig. 5. Examples of head nodes.

Let us give the following notations:

$Dir_M(A)$: moving direction of node A.

$Dir_P(A, B)$: direction vector from node A’s position to node B’s position.

$\theta(U, V)$: included angle (0°~180°) between vectors U and V.

VAG(A, B): visual angle of node B from the viewpoint of the moving direction of node A, which is equal to $\theta(Dir_P(A, B), Dir_M(A))$

MAG(A,B): included angle between the moving directions of nodes A and B, which is equal to θ(Dir_M(A), Dir_M(B)).

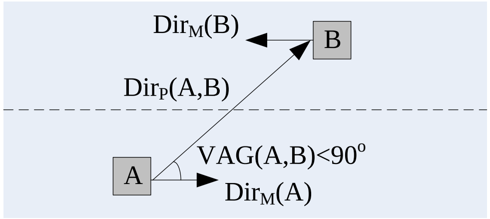
Dist(A, B): Euclidean distance between nodes A and B.

We define the following terms that are related to our previous definition of head nodes and their examples as shown in Fig. 6:

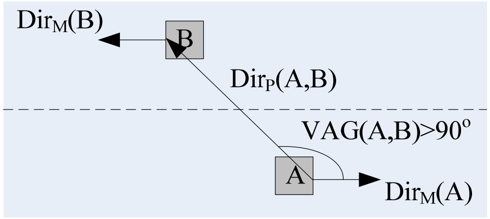
Definition: *B is ahead of A*, *if* 0°≤VAG(A, B)<90°.

Definition: *B is behind of A*, *if* 90°≤VAG(A, B)≤180°.

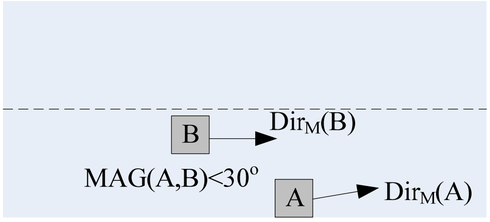
Definition: *A drives the similar direction with B*, *if* 0°≤MAG(A, B)≤30°.



(a) ahead



(b) behind



(c) similar direction of movement

Fig. 6. Scenarios illustrating definitions.

The mission of a receiver node (see procedure ReceiverNode, below) is to parse a received packet and execute procedures ReceiveHello and ReceiveBroadcast when a hello packet and a broadcast packet are received, respectively. In ReceiveHello, the field of recent hello time in the neighbor table is updated accordingly, and a new entry is added if this received packet is from a non-existing node in the table. If an extended hello packet which carries an event list is received, the receiver node compares its own event list with the received one. Any node entering a warning area is detected by finding that this node is missing one emergency event. A scenario is shown in Fig. 7 where both normal nodes A and B receive an extended hello packet from node C and try to compensate node C for the missing event. To prevent redundant broadcast, self-pruning is performed by nodes A and B such that only the closest one (node A) to node C is responsible for the compensation. Moreover, this compensation is performed through local one-hop broadcast.

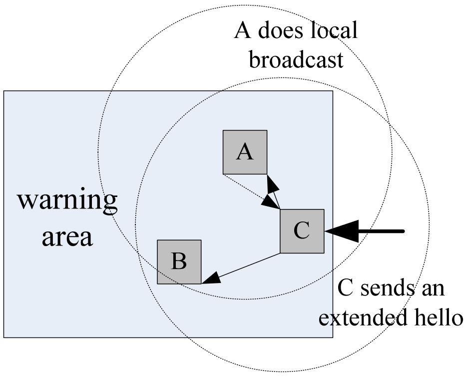


Fig. 7. Local compensation by the closest normal node.

Using the same example, the timing diagram in Fig. 8 shows the different behaviors of node A as a head node or a normal node. Node C at t_1 sends out a basic hello packet which enables node A to identify node C to be a new neighbor. If node A is a head node, node A broadcasts the event quickly, at t_2. Moreover, this broadcast might trigger node C to continue rebroadcasting. However, if node A is a normal node, node A just sends out an extended hello packet at t_3. Meanwhile, node C detects node A to be a new neighbor and sends out an extended hello packet at t_4 as well. As a result, node A finds the missing event from node C and performs a local broadcast at t_5. Node A returns to use the basic hello packet at t_6. Therefore, a head node can broadcast the event more promptly than a normal node but may cause future redundant broadcast or unnecessary rebroadcast. Hence, head nodes only exist when the event propagation is in spreading phase in our scheme.

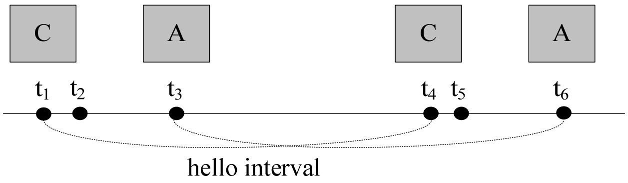


Fig. 8. Difference between normal and head nodes.

In ReceiveBroadcast, we follow the design phases discussed in Section 2 to coordinate multiple receiver nodes. At first, a receiver node cleans up a redundant or invalid broadcast packet (message confirmation phase). If the received packet is not a local broadcast packet, this receiver node waits for a certain time period (waiting delay phase) and then decides whether to rebroadcast the packet or not (rebroadcast decision phase).

We provide two waiting delay functions. The first one is based on the conventional distance function [15] as follows:

$$\text{WT (Waiting Time)} = WT_{\max} \times \left(1 - \frac{D^{\varepsilon}}{R^{\varepsilon}}\right) \ (\varepsilon = 1 \text{ or } 2)$$

D is the distance between a receiver node and a sender node from which the broadcast packet is received. The sender location information has been embedded into the broadcast packet. R is the uniform transmission range of each node in the network. $WT_{\max}$ is set to be twice the average one-hop communication delay.

The second one is based on random time selection between receiver nodes as follows:

$$\text{WT} = WT_{\max} \times \frac{random(0, N_b - 1)}{N_b - 1}$$

N_b is the number of neighbors of a sender node. To get this number at receiver side, we need to add a new field with this number to the hello packet.

In the rebroadcast decision phase, we apply the concept of neighbor elimination as introduced in Section 2. Ideally, those receiver nodes with the same sender node would sequentially rebroadcast the received packet after different waiting times. During the waiting period, a receiver node may listen to ongoing rebroadcasts of the same packet from its neighbors. After the waiting time is over, a receiver node records these rebroadcast nodes and the original sender node in a set RS. A receiver node then need not rebroadcast the packet if all its neighbors have received the packet from some node in RS. However, it is not easy to know if this is the case unless the neighbors report their event tables.

Here, we use a simple hypothesis that a node can successfully receive a packet from a sender node with high probability if this node is within the transmission range of the sender node and there is no shadow effect between them. The shadow effect frequently happens in an urban environment from surrounding buildings or vehicles but is hard to predict in advance without sufficient environmental knowledge. Again, we use a simple method to predict the occurrence of shadowing between two communication nodes if the visual angle (VAG) between them is close to a right angle. This occurs when two nodes are separated by a street corner with buildings. Let us define the following term:

Definition: *Two nodes A and B can successfully exchange data if they are visible to each other. This is, the condition* $Visible(A, B)$ *is true if* $Dist(A, B) \leq R - \Delta$ *and* $(VAG(A, B)$ *or* $VAG(B, A)) \leq 45°$ *or* $\geq 135°$.

In the above definition, Δ is a guard space (pre-configured as one tenth of R) to prevent a sudden disconnection between two nodes that are located on the borders of their communication ranges. This condition of visibility is not always correct. For instance, two nodes moving in perpendicular directions can possibly communicate with each other if one happens to be located at a road junction. Such special cases need more information to judge and are ignored in the paper.

Neighbor elimination is achieved by doing a coverage check with the result further transformed into a rebroadcast probability. In the coverage check, we count the neighbors of a receiver node A that are visible to one of the nodes in set RS collected by node A.

Definition: $Cover(A, RS) = \{v \mid v \in Neighbor(A)$ *and* $Visible(A, s)$ *is true,* $\exists\, s \in RS\}$*, where* $Neighbor(A)$ *is the set of one-hop neighbors of node* A (*including node* A *itself*).

One neighbor in set Cover(A, RS) is supposed to receive the same broadcast packet as node A with a high probability from the same sender node or other rebroadcast nodes. The rebroadcast probability of a receiver node A (P_A) is dependent on how many of its neighbors are in the cover set and is computed as below:

$$P_A = 1 - |\mathrm{Cover(A, RS)}|/|\mathrm{Neighbor(A)}|.$$

This receiver node A will rebroadcast the received packet if a randomly generated number between 0 and 1 is less than P_A. After each packet is sent, we always do a reliability check as mentioned before to increase delivery reliability. Finally, we check whether this receiver node is suitable to be a head node. As a result, a rebroadcast node need not be a head node, but a head node is always a rebroadcast node.

Finally, we show how our proposed scheme solves the problems encountered in Fig. 1. Car A is initially a source node and has neighbors D and E. Car A becomes a head node by broadcasting an emergency event and then continues to be a head node. When car A catches up with car B, car A rebroadcasts the event and stops being a head node. When cars D and E listen to the broadcast from node A, they compute their own rebroadcast probabilities. Car D has a neighbor set {A, E, D}; since all members are covered by A, P_D is zero. Car D decides not to rebroadcast but acts as a head node. Car E has a neighbor set {A, C, D, E, F} with members A, D, and E being covered by A, so $P_E = 2/5$. If car E decides to rebroadcast, cars C and F can listen to the broadcast. However, car E does not meet the requirements to be a head node.

Procedures for each type of node:

```
NormalNode()
Begin
  While (true)
    1) Periodically refresh the event table by removing all invalid events.
    2) Periodically refresh the neighbor table by removing those nodes that have left.
    3) Periodically broadcast a hello packet in basic format and in extended format only when a
       new neighbor is encountered.
    4) If an emergency event $e$ is detected, fork a new process that calls SourceNode($e$).
    5) If a packet $p$ is received, fork a new process that calls ReceiverNode($p$).
```

```
    End while.
End.

SourceNode(Event e)
Begin
    1) Create a new broadcast packet p for the event e by setting the following data fields:
          - Source ID = current node ID
          - Warning time, warning area
          - Sender location = current node location
          - TTL = 256.
    2) Insert this packet p into the event table.
    3) Call HeadNode(p).
    4) Return to the calling process.
End.

HeadNode(Packet p)
Begin
    1) If the broadcast packet p has become invalid, remove the packet from the event table and
         return to the calling process.
    2) If there are any neighbors in the neighbor table,
        2.1) Broadcast this packet and do a reliability check.
        2.2) Check whether to be a head node or not.
               Yes: Go to step 4.
               No: Return to the calling process.
    3) If two successive newly encounter neighbors have the broadcast packet p, end the process.
    4) Otherwise, carry the broadcast packet, wait for a moment and then go to step 1.
End.

ReceiverNode(Packet p)
Begin
    1) If the received packet p is a hello packet, call ReceiveHello(p).
    2) If the received packet p is an emergency broadcast packet, call ReceiveBroadcast(p).
    3) Return to the calling process.
End.

ReceiveHello(Packet p)
Begin
    1) Update the neighbor table and add a new entry for a new neighbor.
    2) If the received packet is an extended hello packet, check whether this neighbor has missed
          any broadcast packets.
         Yes: If the current node is the closest one to this neighbor in the neighborhood,
                locally broadcast the missing packets with TTL = 1.
         No: Return to the calling process.
End.

ReceiveBroadcast(Packet p)
Begin
    1) Check whether the packet p is received for the first time.
         No: Go to step 4.
         Yes: Insert this broadcast packet p into the event table.
```

```
    2) If the packet has become invalid, remove the packet p from the event table and go to step 4.
    3) If the packet is not a local broadcast message,
        3.1) Initiate a waiting timer and after time is up, do a cover check and compute a
             rebroadcast probability.
        3.2) Randomly generate a number between 0 and 1 and rebroadcast the packet with updated
             header fields if the generated number is less than the rebroadcast probability. Do a
             reliability check when rebroadcast is performed.
        3.3) Check whether to be a head node or not.
             Yes: Call HeadNode(p).
             No: Go to step 4.
    4) Return to the calling process.
End.
```

4. Performance Evaluation

To evaluate the performance of our proposed broadcast scheme, we carried out simulations using NS-2.33 [25]. To simplify our implementation, we observed the result of a single-event propagation. In other words, only one event is triggered in the whole network. The cost metrics are listed as below:

Average delay time: average elapsed time from the moment when a vehicle enters into a warning area and remains there to the moment when the vehicle receives an emergency broadcast packet.

Rebroadcast times: total number of times that an emergency broadcast packet is rebroadcast after it is issued from a source node.

Delivery ratio: ratio of the number of vehicles that correctly receive an emergency broadcast packet to the number of vehicles that have ever entered into a warning area during the warning time.

Control overhead: ratio of the total sizes of hello packets exchanged to the number of vehicles that correctly receive an emergency broadcast packet.

4.1 Simulation Model

We consider a real street environment which is imported from the TIGER (Topologically Integrated Geographic Encoding and Referencing) database [26]. A city street map of size 2000m×2000m is used as shown in Fig. 9. Under the street model, vehicles are generated and their moving patterns are controlled by the tool VanetMobiSim [27] which supports multi-lane roads, differentiated speed constraints, and traffic signs at intersections. In our configuration, each road has two lanes (forward and reverse) and there are totally 20 traffic lights. Each vehicle picks a destination on the road and moves toward the destination along the shortest driving path with a speed range from 20 km/hr to 60 km/hr. Once the vehicle reaches the destination, it picks another destination randomly after a pause time from 5s to 30s. For each simulation run, one vehicle is randomly selected as an event source node. The warning area is a circle centered at the current location of the source node. The warning area is fixed but the source node can continue moving. Each cost result is computed through the average of ten simulation runs. The default parameter settings in our simulation are listed in Table 2.

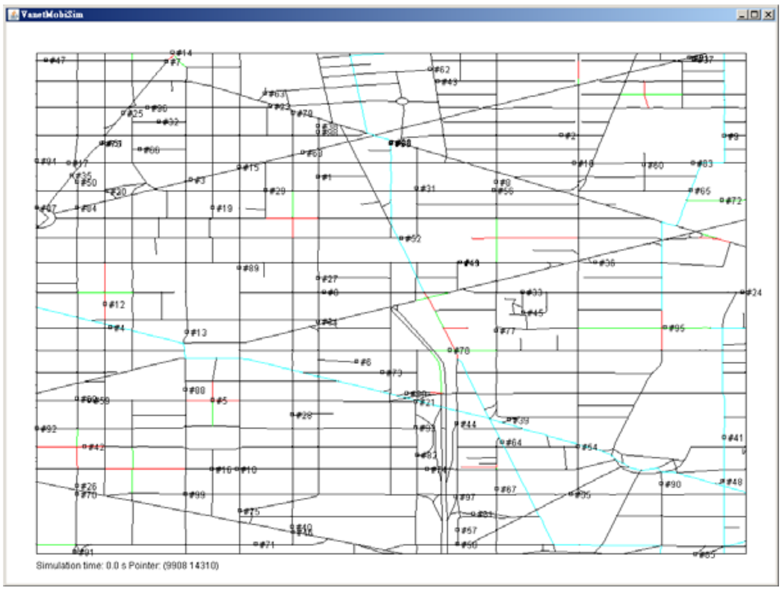


Fig. 9. Simulation environment.

Table 2. Parameter settings.

Parameter	Value
Transmission radius	100 m
MAC protocol	802.11p
Physical propagation model	Two-ray ground
Number of nodes	250
Vehicle speed	20 km/hr ~ 60 km/hr
Radius of a warning circle	200 m
Warning time	180 s
Hello interval	1 s
WTmax	4 ms
Waiting delay	Distance (ε=2)
Simulation time	240 s

We compare our proposed scheme with those schemes that provide complete broadcast solutions. From the viewpoint of implementation cost, schemes EDB, DP-IB, UMB, and AMB are not practical due to the requirements of roadside units or map information. Therefore, the target schemes to be compared are WPP, SPP (S1P is a special case of SPP), and RBM. In WPP, the waiting delay is fixed with time $WT_{\max}$ and the rebroadcast probability is D/R. In SPP, the waiting delay is computed by the following formula:

$$N_s \times (1 - \left\lceil \frac{\min(D,R)}{R} \right\rceil) \times WT_{\max} .$$

N_s is the maximum number of sectors and is set to 4 in the simulation. The rebroadcast probability of SPP is assigned a fixed value.

In RBM, the waiting delay uses the same distance function introduced in the previous section. Moreover, the neighboring nodes of the current broadcast node are divided into six directions according to their moving directions. Nodes moving in each direction compete internally to be rebroadcast nodes. Such a design can solve the intersection problem. Neither WPP nor SPP provide the carry and forward mechanism, but RBM does.

4.2 Simulation Results

A. Effect of waiting delay functions

We compare the performance using distance-based and random-based delay functions on our WWB broadcast scheme. These two functions are labeled as Distance-ε-$WT_{\max}$ and Random-$WT_{\max}$ in the experiment by setting different ε and $WT_{\max}$ values. In Fig. 10, we find that the random delay function achieves the highest delivery ratio at each test point. Furthermore, $WT_{\max}$ = 4 ms is preferred as the node density is high. Distance-2-2ms and Distance-1-4ms almost have the same low delivery ratio.

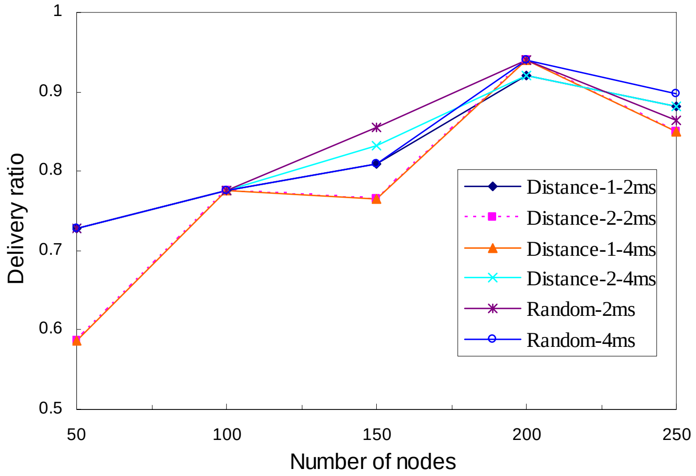


Fig. 10. Waiting delay functions: delivery ratio.

In Fig. 11, we find that delay functions with larger ε and $WT_{\max}$ values generate a lower number of rebroadcast times. Distance-1-2ms and Random-4ms respectively generate the most and the least rebroadcast times on average. In Fig. 12, Distance-2-2ms and Distance-1-2ms respectively have the lowest and the highest delay time on average. In conclusion, Random-4ms is preferred for having a high delivery ratio and Distance-2-2ms is preferred for having a low delay time.

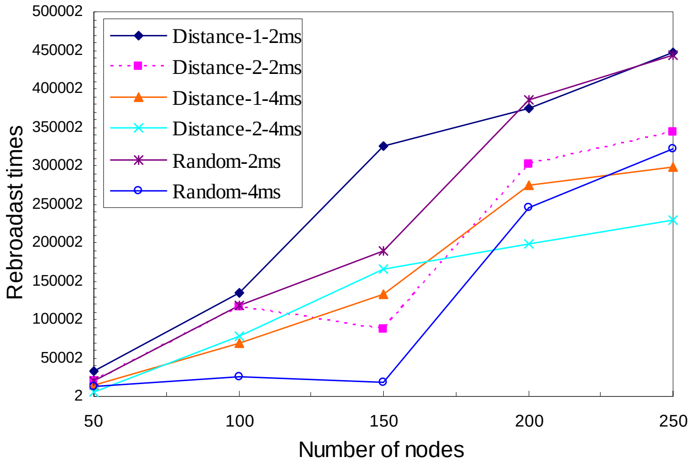


Fig. 11. Waiting delay functions: rebroadcast times.

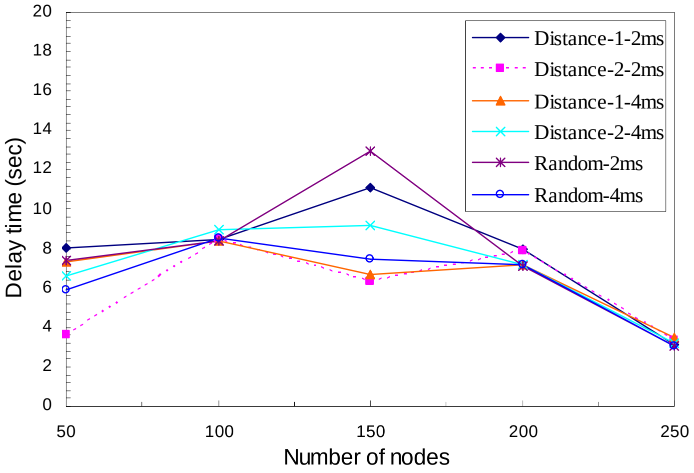


Fig. 12. Waiting delay functions: delay time.

From a packet collision point of view, a good delay function should disperse the time points of rebroadcasting such that more later rebroadcasts are prevented by former ones. Therefore, a second-order distance function (with ε=2) is certainly better than a one-order function (with ε=1). A larger $WT_{\max}$ value is preferred as well and this result is more apparent under dense vehicle environments. To achieve a high delivery ratio, one method is to increase broadcast coverage. However, neighboring nodes self-preventing rebroadcast would possibly cause some broadcast holes where some nodes are not under broadcast coverage. Using the distance delay function, outer nodes always prevent inner nodes from rebroadcast. However, this situation is more balanced using the random delay function, and hence the broadcast hole problem is less serious. Therefore, we observe high delivery ratios with random delay functions. On the other hand, giving high precedence of rebroadcast to outer nodes can speed up broadcast propagation. Setting a small $WT_{\max}$ value for the outer nodes has the same effect as well. That is why the distance delay function causes a lower delay time than the random one, and why 2 ms is more suitable than 4 ms as the distance delay function in delay time.

B. Effect of warning time and warning area

In this experiment, we observe the performance of WWB with different lengths of warning time and different sizes (radii) of warning area. The size of warning area has a great influence on delivery ratio as shown in Fig. 13. However, the length of warning time has less influence. This is because the carry and forward mechanism plays an important role here. Nodes in the warning area can receive the warning message when they encounter any head node or normal node. If the warning area is small, encounters with other nodes happen with high probability. However, this encounter probability is small if the warning area is too large.

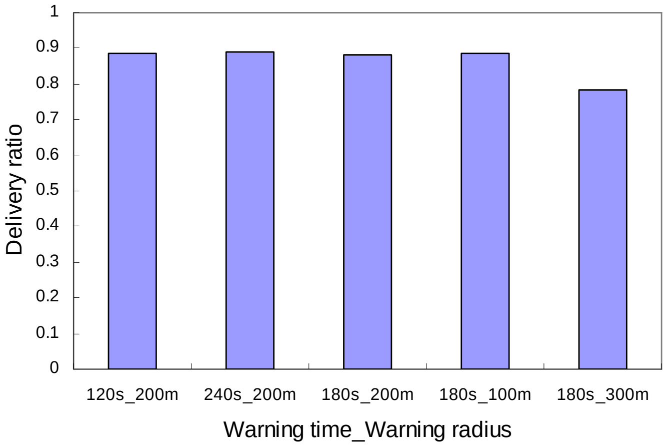


Fig. 13. Waning time/area: delivery ratio.

In Fig. 14, the rebroadcast times increase as the warning time or the warning area gets larger. This is because more nodes are involved in message spreading. As can be seen in Fig. 15, the delay time is increased as the warning area is increased but is decreased as the warning time is increased. In a large area, it takes time for a node to encounter another node, so this prolongs the delay time. With the passage of time, most nodes that have already received the warning message are leaving the warning area. Any nodes coming into the warning area later will easily meet these nodes and get the warning message. Therefore, the delay time shortens.

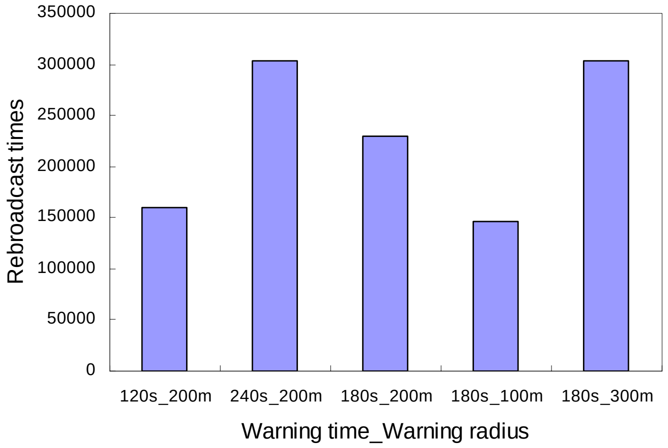


Fig. 14. Waning time/area: rebroadcast times.

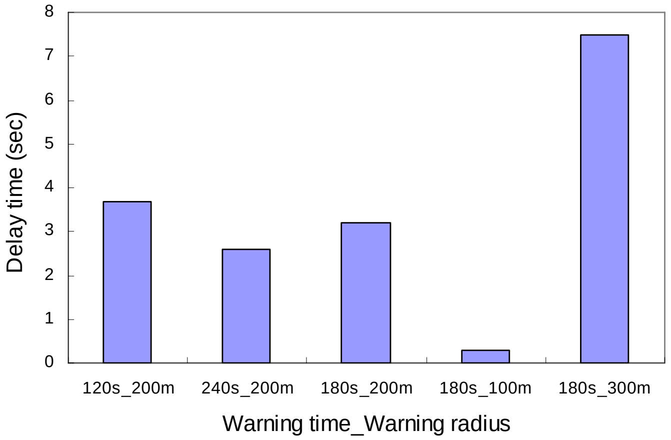


Fig. 15. Warning time/area: delay time.

C. Comparison of different schemes

WWB greatly outperforms the other schemes in delivery ratio, as shown in Fig. 16, for two main reasons. The first is because carry and forward largely overcomes connection hole problems. In our testing environment, there are many connection holes. RBM using the same mechanism shows some improvement too. Other schemes without this mechanism perform very poorly. The second one is local broadcast by normal nodes during the warning time. Any nodes entering a warning area can be notified in WWB. However, other broadcast schemes just emphasize the initial event propagation to all nodes connected at the moment when an event is triggered. After this event propagation, a node arriving in a warning area later cannot receive the event message.

The delivery ratio of WWB increases as the number of nodes increases, since the network topology becomes more connected and more nodes can carry broadcast packets. The other schemes mostly present an up-and-down phenomenon in delivery ratio. The increase is due to less connection holes and the decrease is due to more nodes arriving in the warning area without being notified. WPP and SPP series with different rebroadcast probabilities perform similarly when the number of nodes is less than 200.

To support the warning time service, we provide an extension to RBM (labeled as RBM*) by allowing the source node to rebroadcast the event periodically (an idea introduced in ODAM). With an interval time of 1s, the delivery ratio is indeed increased but is still lower than our approach. Hence, simple periodic rebroadcast is not necessarily a good solution.

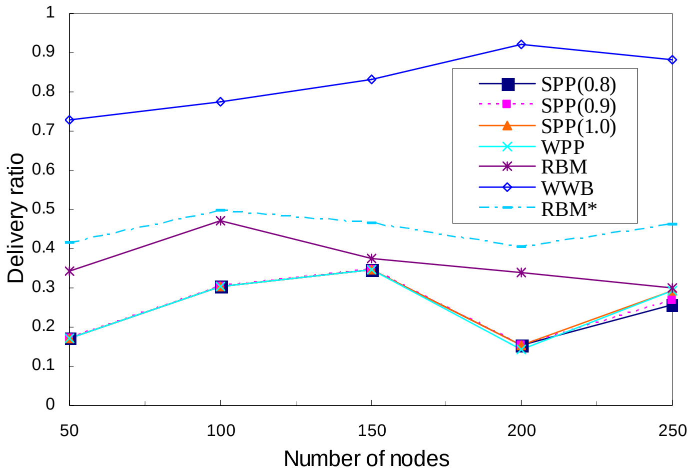


Fig. 16. Delivery ratio comparison.

The result of rebroadcast times is demonstrated in Fig. 17. This figure shows a trend similar to the previous figure. WWB generates many rebroadcasts for the aforementioned reasons. These rebroadcasts are not redundant but increase delivery ratio. Hence, using carry and forward will increase rebroadcast times as in the case of RBM. The rebroadcast times of WPP and SPP are very low due to their low delivery ratios.

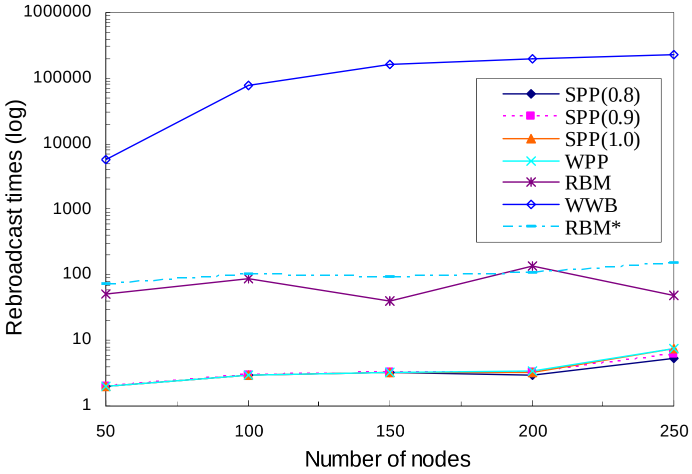


Fig. 17. Rebroadcast time comparison.

The result of delay time is shown in Fig. 18. This metric indicates how fast each node in a warning area is informed of an event. WWB has the highest value, due to using carry and forward. The radio propagation speed is extremely fast compared to the moving speed of a vehicle. Therefore, WPP and SPP, in which any communications are through radio, have very low delay time (less than 1s). The delay time of WWB is reduced as the number of nodes is increased, since any two nodes in a dense environment can encounter each other in a short time. RBM has less delay time than WWB, because nodes entering a warning area are not handled by carry and forward.

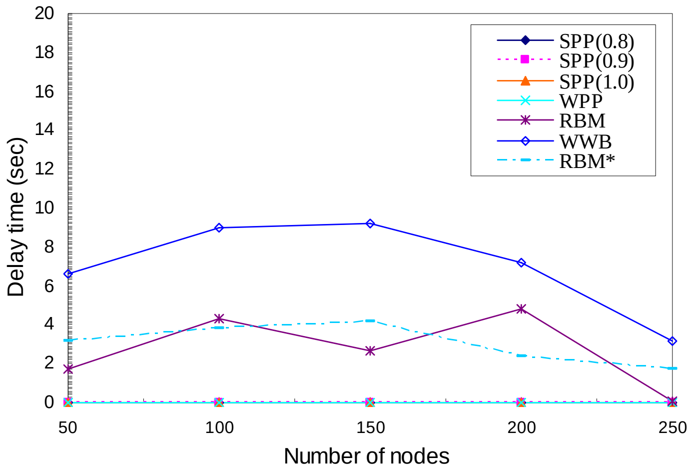


Fig. 18. Delay time comparison.

Finally, we compare the control overhead as shown in Fig. 19. The control overhead results from hello packets. WPP and SPP which use sender/receiver knowledge on rebroadcast decision do not generate any hello packets and hence the control overhead is zero. RBM uses sender/receiver knowledge but involves hello packets in dealing with connection hole problems. WWB is based on neighbor knowledge that generates periodic basic hello packets and occasional extended hello packets. For example, 12% control overhead is from extended hello packets when the number of nodes is 200. As can be seen, WWB has a higher control overhead than RBM.

In conclusion, WWB provides an excellent delivery ratio compared to other schemes. Though the delay time may be long due to carry and forward, this result is acceptable for several reasons. When the node density is very high, less communication takes place through carry and forward and hence the delay time of WWB improves. When the node density is very low, emergency events are less dangerous and late warning will result in less sudden braking. The only weakness of WWB is control overhead. However, the current 802.11p specifications support control channels separated from data channels and periodic beaconing. This control overhead is affordable.

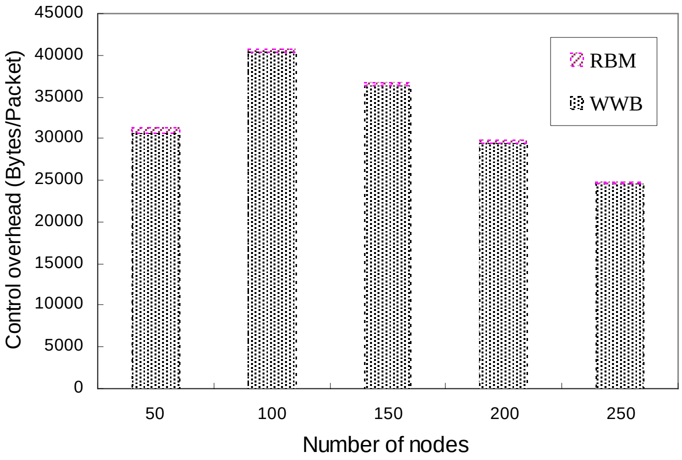


Fig. 19. Control overhead comparison.

5. Conclusions

The dissemination of safety-related messages is an important application in a vehicular network environment. Highly dynamic features make vehicular networks unstable. Traditional broadcast mechanisms designed for MANET become inappropriate in this kind of environment. Design of an efficient and high reliable broadcast scheme is a critical issue for improving traffic safety.

We identify four problems that should be handled by broadcast design in VANET: broadcast storm, connection hole, building shadow, and intersection problems. Most existing broadcast schemes do not provide full solutions to these problems and more importantly these schemes also do not provide full support for warning services with specified area and duration. Our proposed broadcast scheme is free from these limits. By simulating water wave propagation and through exploitation of node role transition, the proposed scheme is easy to understand and implement. Given the results of our experiments, our proposed scheme has an excellent performance to delivery ratio. The delay time is dominated by the utilization rate of carry and forward.

One issue that would need to be studied in the future is differentiation of emergency messages with various priorities, with further modification of the corresponding broadcast scheme. To prevent false warnings from malicious sources, traditional security issues such as integrity and non-repudiation should be considered as well.

References

[1] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives," *in Proc. Intl. Conf. on ITS Telecommunications (ITST),* pp. 761-766, Jun. 2006.

[2] "Car2Car Communication Consortium," http://www.car-2-car.org.

[3] "NOW Project," http://www.network-on-wheels.de/.

[4] "FleetNet Project, Internet on Road," http://www.informatik.uni-mannheim.de/pi4/projects/FleetNet.

[5] "VICS (Vehicle Information Communication System)," http://www.vics.or.jp/english/index.html.

[6] Y. Toor, P. Muhlethaler, A. Laouiti, "Vehicle Ad Hoc Networks: Applications and Related Technical Issues," *IEEE Communication Surveys and Tutorials*, vol. 10, no. 3, pp. 74-88, Jul./Sept. 2008.

[7] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116-126, Sept. 1997.

[8] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *in Proc. ACM MOBICOM*, pp. 151-162, 1999.

[9] C. Y. Yang and S. C. Lo, "Street Broadcast with Smart Relay for Emergency Messages in VANET," *in Proc. IEEE AINA Workshop*, May 2010.

[10] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *in Proc. ACM MOBIHOC*, pp. 194-205, 2002.

[11] Stojmenovic, M. Seddigh, and J. Zunic, "Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks, " *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14-25, Jan. 2002.

[12] Stojmenovic and J. Wu, "Broadcasting and Activity-Scheduling in Ad Hoc Networks, *in Ad Hoc Networking*, IEEE Press Wiley, 2003.

[13] J. Davis, A. Fagg, and B. Levine, "Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks," *in Proc. Intl. Symp. on Wearable Computing*, Oct. 2001.

[14] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast Storm Mitigation Techniques in Vehicular Ad Hoc Networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 84-94, Dec. 2007.

[15] Benslimane, "Optimized Dissemination of Alarm Messages in Vehicular Ad-Hoc Networks," *in Proc. High Speed Networks and Multimedia Communications*, pp. 655-666, 2004.

[16] D. Li, H. Huang, X. Li, M. Li, and F. Tang, "A Distance-Based Directional Broadcast Protocol for Urban Vehicular Ad Hoc Network," *in Proc. Intl. Conf. on Wireless Communications, Networking and Mobile Computing*, pp. 1520-1523, 2007.

[17] J. Zhao, Y. Zhang, G. Cao, "Data Pouring and Buffering on the Road: A New Data Dissemination Paradigm for Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3266-3277, Nov. 2007.

[18] H. Jiang, H. Guo, and L. Chen, "Reliable and Efficient Alarm Message Routing in VANET," *in Proc. IEEE ICDCS Workshop*, pp. 186-191, 2008.

[19] Z. Shi, F. Liu, and S. Xu, "Novel Relay Scheme Based on Traffic Type in Vehicular Networks," *in Proc. Intelligent Computation Technology and Automation* (*ICICTA*), pp. 392-397, 2008.

[20] S. Khakbaz and M. Fathy, "A Reliable Broadcast Method for Vehicular Ad Hoc Networks Considering Fragmentation and Intersection Problems," *in Proc. IEEE Intl. Conf. on Next Genartion Mobile Applications, Services, and Technologies*, pp. 379-384, 2008.

[21] M. T. Sun, W. C. Feng, T. H. Lai, K. Yamada, and H. Okada, “GPS-Based Message Broadcasting for Inter-Vehicle Communication,” *in Proc. Intl. Conf. on Parallel Processing*, pp. 279-286, 2000.

[22] M. Koubek, S. Rea, and D. Pesch, “A Novel Reactive Routing Protocol for Applications in Vehicular Environments,” *in Proc. Intl. Symp. on Wilreess Personal Multimedia Communications*, 2008.

[23] E. Fasolo, A. Zanella, and M. Zorzi, “An Effective Broadcast Scheme for Alert Message Propagation in Vehicular Ad hoc Networks,” *in Proc. IEEE Intl. Conf. on Communications*, pp. 3960-3965, 2006.

[24] G. Kokmaz, E. Ekici, F. Ozguner, "Black-Burst-Based Multihop Broadcast Protocols for Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, vo. 5, pp. 3159-3167, Sept. 2007.

[25] NS-2.33, “The Network Simulator,” http://www.isi.edu/nsnam/ns/

[26] TIGER, “Topologically Integrated Geographic Encoding and Referencing,” http://www.census.gov/geo/www/tiger.

[27] VanetMobiSim, http://vanet.eurecom.fr/