Computer Networks 55 (2011) 2592–2607

Contents lists available at ScienceDirect

Computer Networks

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

On the performances of forwarding multihop unicast traffic in WBSS-based 802.11(p)/1609 networks

Shie-Yuan Wang *, Chih-Che Lin, Wei-Jyun Hong, Kuang-Che Liu

Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 31 July 2010
Received in revised form 17 March 2011
Accepted 3 May 2011
Available online 13 May 2011
Responsible Editor: J. Misic

Keywords:
Inter-vehicle communications
802.11(p)
IEEE 1609
Multihop forwarding
WBSS

ABSTRACT

The IEEE 802.11(p)/1609 network is a promising candidate for future vehicular communication networks. In this new network, the operation of a new Wave Basic Service set (WBSS) is defined for vehicular environments. Due to the deployment cost consideration, roadside units (RSUs) in such networks are usually installed only at hot spots and intersections, causing the service coverage of RSUs to be discontinuous. To overcome this problem, multihop data forwarding among vehicles can be used to extend the service coverage of RSUs.

The multihop geocasting problem has been studied and addressed in our previous work [1], where we proposed a receiver-centric WBSS geocasting scheme to increase multihop geocasting performances in WBSS-based networks. However, multihop unicasting, a more common facility in many network applications and services, was not dealt with in [1].

In this paper, we propose a new Receiver-centric WBSS Forwarding Scheme for Unicast traffic (called RWFS-U), which uses a prioritized receiver-centric WBSS creation design to efficiently support unicast data forwarding in WBSS-based networks. Extensive performance evaluations using both analyses and simulations are presented to understand the effects of the designs of WBSS forwarding schemes on the goodputs of end-to-end flows. Our results show that RWFS-U can significantly outperform the traditional sender-centric schemes on multihop unicasting in WBSS-based networks.

1. Introduction

Vehicular networks have obtained great attention in recent years due to the challenges resulting from its extremely dynamic nature. The IEEE 802.11(p) specification [2], which amends the IEEE 802.11-2007 standard [3], is currently under-development for this type of networks. It defines a new MAC-layer operational mode for Wireless Accesses in Vehicular Environments (called the WAVE mode). The WAVE mode so far defines two basic service sets. One is the WAVE Basic Service Set (WBSS), which comprises a provider and several users and is mainly used for Roadside Units (RSUs) [4] to communicate with Onboard Units (OBUs) [4]. Data communication is only allowed between a provider and its users. The WBSS defined in the 802.11(p) network is analogous to the infrastructure BSS defined in the traditional 802.11(a/b/g) network. The key difference between them is that, after listening to a beacon message of a WBSS, a new user can directly join the WBSS without performing the authentication and association procedures. The details of a WBSS are explained in Section 2.

The other type of BSS is the WAVE Independent Basic Service Set (WIBSS), which comprises multiple peer users. Data communication is allowed between any pair of users. The WIBSS in the 802.11(p) network is analogous to the Independent Basic Service Set (IBSS) defined in the traditional 802.11(a/b/g) network. The main difference between them is that the former explicitly excludes the use of

* Corresponding author.
E-mail address: shieyuan@csie.nctu.edu.tw (S.-Y. Wang).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.05.003

beacon messages in its operation while the latter use them to synchronize the clocks of nodes.

As shown in Fig. 1 [5], the IEEE 802.11(p) specification and the IEEE 1609 standard family [4,6–9] co-define a complete protocol suite for vehicular networks (denoted as the IEEE 802.11(p)/1609 network). The Wave Management Entity (WME) is responsible for creating/destroying WBSSs and determining which WBSS the node should join. The IEEE 802.11(p)/1609 network supports the TCP/UDP/IP protocol suite and a new WAVE-mode short message protocol (WSMP). The former is used to serve traditional network applications while the latter is used to disseminate small packets that carry emergent safety and traffic information.

The IEEE 802.11(p)/1609 network manages link bandwidth in a combined FDMA/TDMA manner. As shown in Fig. 2 [5], in this network the available frequency spectrum is divided into one control channel (CCH) and six service channels (SCHs). The CCH is used by nodes to exchange their network control messages while SCHs are used by nodes to exchange their data. WAVE-mode short messages (WSMs) can be transmitted on both CCH and SCHs. The link bandwidth of each of these channels is further divided into transmission cycles on the time axis, each comprising a control frame and a service frame. In a transmission cycle, the control frame must be on CCH whereas the service frame can be on any SCH.

Due to the deployment cost consideration, RSUs are usually installed only at hot spots and intersections. In this condition, an OBU may not be able to directly connect to a RSU at all time. To overcome this discontinuous coverage problem, multihop data forwarding among vehicles can be used to help extend the coverage of RSUs. That is, a vehicle can use multihop forwarding to exchange its packets with a RSU that is not within its radio range.

Multihop data forwarding can be achieved over a WIBSS. This approach is very similar to using multihop data forwarding over a traditional IBSS, which has been extensively studied in the literature. In an IBSS/WIBSS, however, every node normally needs to operate on the same channel. As a result, the bandwidth of multiple SCHs cannot be used at the same time unless a complicated protocol is used. Multihop data forwarding can also be achieved by using WSMP, which allows nodes to transmit data using WSMs without forming a basic service set in advance. However, normally WSMP is used for transmitting small packets carrying emergent information rather than for transmitting large normal data packets. In addition, to use WSMP, every node needs to operate on the same channel, which incurs the same problem with WIBSS.

These observations motivated us to design a solution that can easily deploy multihop forwarding in WBSS-based networks. Our proposed solution has two objectives: (1) to efficiently deploy multihop data forwarding in WBSS-based networks and (2) to spread traffic generated by different nodes over different service channels to increase total network capacity. In our previous work [1], we proposed a Receiver-centric WBSS Forwarding Scheme (called RWFS) to enhance the multihop forwarding performances of WBSS-based vehicular networks. In that paper, performance results of RWFS in various scenarios were compared with those of a typical Sender-centric WBSS Forwarding Scheme (called SWFS).

The RWFS proposed in [1], however, is mainly designed for multihop geocasting traffic. As a result, it cannot serve unicast traffic. However, unicast is commonly used in many existing network applications and services. This motivated us to revise the SWFS and RWFS proposed in [1] so that they are capable of performing multihop unicasting and cooperating with unicast ad hoc routing protocols (such as DSDV [10] and AODV [11]). In this paper, we propose a version of RWFS for relaying multihop unicast traffic (called RWFS-U). RWFS-U solves several issues on relaying multihop unicast data in an IEEE 802.11(p)/1609 network and enhances the multihop forwarding performance in such a new network. The performances of RWFS-U are compared with those of a version of SWFS for relaying multihop unicast traffic (called SWFS-U). Extensive performance comparisons between these two schemes on different topologies are conducted using both analytical and simulation approaches.

The remainder of this paper is organized as follows. In Section 2, we introduce the operation of a WBSS and its problem with multihop unicast data forwarding. In Section

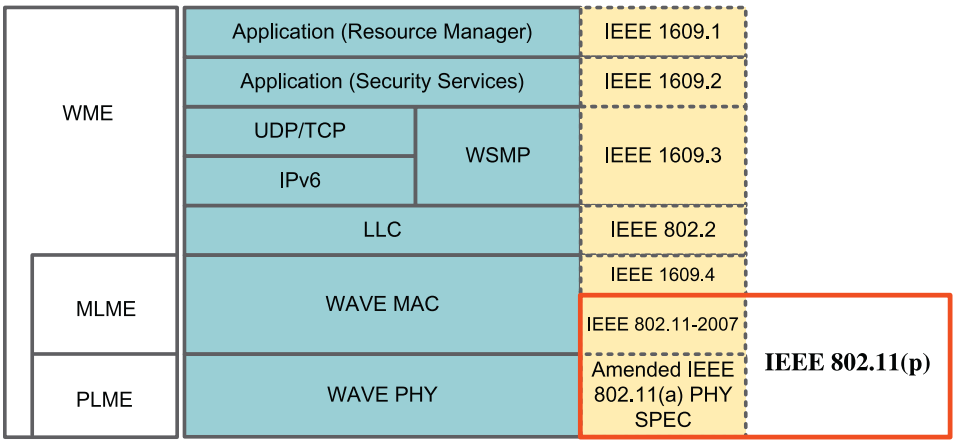


Fig. 1. The protocol stack of an 802.11(p)/1609 network node.

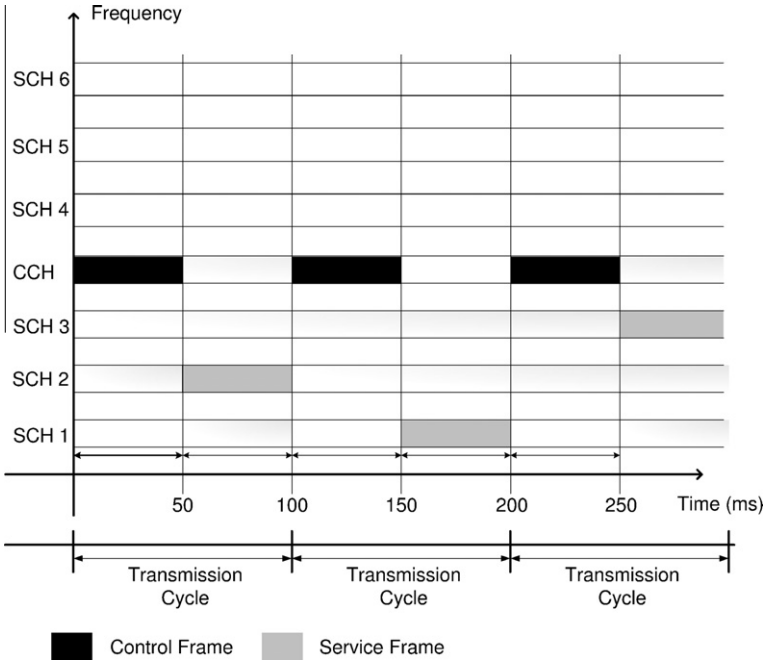


Fig. 2. Example of the frequency division in an 802.11(p)/1609 network.

3, we discuss the issue of control message dissemination in WBSS-based networks for routing protocols. In Section 4.1, we explain the design of the basic sender-centric WBSS-based forwarding scheme (SWFS-U) for unicast traffic, which is based on a typical sender-centric WBSS-creating scheme, In Section 4.2, we propose a receiver-centric WBSS-creating scheme to solve the problems depicted in Section 2. Based on it, a receiver-centric WBSS-based forwarding scheme for unicast traffic is proposed (called RWFS-U). The performance evaluation of SWFS-U and RWFS-U using both analytical modeling and simulations are presented in Section 5. Finally, we survey related work in Section 6 and conclude this paper in Section 7.

2. The operation and problem of a WBSS

An 802.11(p)/1609 node should first operate on CCH to gather necessary network information after it joins the network. Data packet transmissions are only allowed to occur within a WBSS. A node that creates a WBSS is called a WBSS provider and nodes that join a WBSS are called WBSS users. To establish a WBSS, a WBSS provider needs to broadcast beacon frames that contain a WAVE Service Advertisement (WSA) message for this WBSS on CCH. A WSA message contains the operational information of its WBSS (e.g., the ID of the WBSS and the SCH that will be used by this WBSS). A node should monitor all WSA messages on CCH on control frames to know the existence and the operational information of available WBSSs. After obtaining the operational information of a WBSS, a node can join the WBSS by switching its channel to the SCH used by the WBSS on service frames.

A WBSS user does not need perform the authentication and association procedures to join a WBSS. The main reason is that, in a highly-mobile environment such as a vehicular network, wireless link connectivity among vehicles is very fragile and has very short lifetime. By using this design, a vehicle can quickly utilize the bandwidth of a WBSS after detecting its existence. Because a WBSS provider may change the operational parameters of its WBSS, a WBSS user should switch back to CCH constantly to learn the latest information about its WBSS.

The communication in a WBSS is carried out in a one-hop manner, i.e., data exchanges are only allowed between a WBSS user and the WBSS provider. According to [4], "WAVE devices take the role of either provider or user on a given service; this is determined by the role chosen by the application operating through the device." and "A device may change roles as it participates on different services." These statements mean that *a node cannot simultaneously create its own WBSS and join other node's WBSS on a given service at the same time.* Such a requirement is reasonable for a multi-channel network such as an 802.11(p)/1609 network. Without this requirement, a node may join a WBSS and create its own WBSS for the same service at the same time. In this condition, if the SCHs used by these WBSSs differ, it is obvious that the node cannot simultaneously participate in the both WBSSs.

The same dilemma can also occur when the two WBSSs belong to different services. In such a situation, this dilemma can be solved by associating different services with different priorities and let the node determine which SCH to switch to based on their priorities. However, if the two WBSSs belong to the same service, such a dilemma cannot be easily solved based on priority.

As can be expected, multihop data forwarding is certainly a service to upper-layer applications. Because the standard does not allow a node to join a WBSS (to be a WBSS user) and create a WBSS (to be a WBSS provider)

at the same time for the same service, supporting multihop data forwarding over WBSS-based vehicular networks is a problem. This is because in a general multihop vehicular network some nodes will inevitably need to be both a provider and a user at the same time to forward packets. With this constraint, the sender-centric WBSS-creating scheme (described in the standard) is inefficient for multihop data forwarding. To address this problem, in this paper we propose a receiver-centric WBSS-creating scheme and, based on it, propose a receiver-centric WBSS-based forwarding scheme (i.e., RWFS-U) for unicast traffic.

3. Dissemination of routing control messages

Providing a facility for routing protocols to broadcast their control messages is the first issue to be solved when one tries to deploy multihop forwarding in WBSS-based networks. Most of unicast ad hoc routing protocols need a facility for a node to disseminate (usually broadcast) its locally gathered information to its neighboring nodes. Such dissemination is usually cost free for mobile ad hoc networks (MANETs) due to the broadcast nature of single-channel radios. However, because in WBSS-based vehicular networks OBUs operate on multi-channel radios, it is not guaranteed that all neighboring OBUs are operating on the same channel when an OBU is broadcasting/transmitting its local information. Worse yet, since WBSSs exclude the use of the authorization and authentication procedures, the sending node (provider) does not know whether its neighboring OBUs have joined its WBSS and are ready for receiving its messages before it transmits these messages. This problem is very critical to unicast ad hoc routing protocols as the inconsistency of nodes' local information can cause nodes to generate inconsistent routes and routing loops.

To solve this problem, in our proposal each OBU uses WSMs to broadcast its local information for the upper-layer routing protocols. More specifically, when the WME [8] detects that the upper-layer routing protocol generates an outgoing message, it encapsulates this message into a WSM with the WSM version set to 2 (which denotes "WSM_UNICAST_CTL_MSG") and broadcasts such a special WSM on control frames on CCH. When a WME receives a WSM with its version set to WSM_UNICAST_CTL_MSG, it de-capsulates the special WSM and passes the payload (i.e., the routing control message) to the upper-layer routing protocol. Because each OBU is required to periodically listen to CCH on control frames, such a design enables the routing protocols running on different nodes to exchange their routing information and maintain correct routes.

Several routing protocols need to use unicast control messages to operate correctly. For example, the AODV protocol needs to use unicast RREP messages to determine routes. One way to transmit these unicast control messages is to transmit them as unicast data packets using the proposed SWFS-U or RWFS-U approach. An alternative is to transmit them using WSMs as described above. The former approach consumes less bandwidth on CCH. However, it incurs longer end-to-end delays for these unicast control messages because on each hop the two operations of joining another node's WBSS and creating its own WBSS cannot occur on the same service frame. As a result, a unicast control message needs to wait the interval of a transmission cycle (consisting of a control frame and a service frame) before it can arrive at the next-hop node. Thus, the time required to initiate a route from a source node to a destination node in the AODV protocol is $(T_1 + N_{hop} \cdot (T_{cf} + T_{sf}))$, where T_1 denotes the time required by the RREQ broadcast packet to reach the destination node by using WSMs on each hop (thus without the need to create and join WBSSs on each hop). N_{hop} denotes the number of hops between the source and the destination node; T_{cf} and T_{sf} denote the intervals of a control frame and a service frame, respectively. Because the broadcast of WSMs does not need involve the creations and joins of WBSS, different nodes' WSMs can be broadcast on the same control frame in a short period of time. This reason makes T_1 much less than $N_{hop} \cdot T_{cf}$.

In contrast, the approach of using WSMs for transmitting unicast control message will add minimum end-to-end delay to these messages but at the cost of consuming a small amount of bandwidth on control frames. In this paper, we use this approach to forward unicast routing control messages because the focus of this paper is on studying whether sender-based WBSSs or receiver-based WBSSs are more efficient for serving multihop traffic rather than the effects of the two approaches on transmitting unicast control messages. However, to be complete, in Appendix A we present performance comparison results of the DSDV and AODV protocols under the two approaches.

4. Design of the SWFS-U and RWFS-U schemes

4.1. The basic SWFS-U

Fig. 4 shows the main concept of the SWFS-U scheme. Using SWFS-U, a transmitting node should create a WBSS before transmitting its data. After broadcasting a WSA message for the WBSS that it creates (on a control frame on CCH), the transmitting node assumes that the intended receiving node will join its WBSS (i.e., the receiving node will switch itself to the SCH used by this WBSS) on next service frame to receive its data. Thus, when next service frame arrives, the transmitting node will start sending the intended receiving node its unicast data packets. In a multihop application scenario, the intended receiving node is usually the next-hop node indicated by the upper-layer routing protocol.

For unicast data transmissions, the 802.11(p) MAC can detect a failed transmission when it receives no corresponding MAC-layer ACK frame for this packet. Upon receiving a data packet, a receiving node should determine whether it is the destination of the received packet. If it is, it delivers the payloads of these packets to upper-layer protocols/applications. If it is not the destination node of the received packet but the next-hop node specified in the 802.11 MAC header, the receiving node should enquire the routing protocol about the next-hop node of this packet. It then performs the above process to forward this pack-

et. In case the receiving node is neither the destination node nor the next-hop node of this received packet, it simply discards the received packet. This process is repeated until the data packet finally arrives at its destination node.

4.2. The proposed RWFS-U

The SWFS-U scheme operates based on the following assumption: *After a WBSS is created, at least one neighboring node will join it*. However, this assumption is not always true. For example, as shown in Fig. 3, consider an 802.11(p)/1609 network comprising four nodes, A, B, C, and D, that can listen to each other. Suppose that nodes A and B have data to send and create their respective WBSSs using different SCHs for better channel utilization. Due to the use of different SCHs, other nodes (such as nodes C and D), cannot join both of the WBSSs at the same time. Without coordination, nodes C and D may choose the same WBSS (e.g., node A's WBSS) to join. In this condition, no nodes will join node B's WBSS to receive and forward its packets. Although node B can detect transmission failures by receiving no MAC-layer ACK frames on the next service frame, it will waste the bandwidth of that service frame.

From these observations, we concluded that a *sender-centric* WBSS creating scheme (such as the SWFS-U scheme) cannot efficiently forward data due to two reasons. First, a WBSS provider cannot know whether a WBSS user has joined its WBSS before transmitting its packets. Second, it may happen that most WBSS users choose to join a particular WBSS, causing some WBSSs to be starved (i.e. no WBSS users join them to receive/forward their data packets). To solve these problems, in this paper we first propose a receiver-centric WBSS Creating Scheme (denoted as RWCS) and then, based on RWCS, we propose a Receiver-centric WBSS Forwarding Scheme for Unicast traffic (denoted as RWFS-U) to efficiently forward unicast packets over multiple hops in WBSS-based vehicular networks.

RWCS does not need modify the format of the WSA message and is easy to implement in the current 802.16(p)/1609 network. The main concept of RWFS-U is shown in Fig. 5. Using RWFS-U, a WBSS is created by the receiving node rather than the transmitting node. Suppose

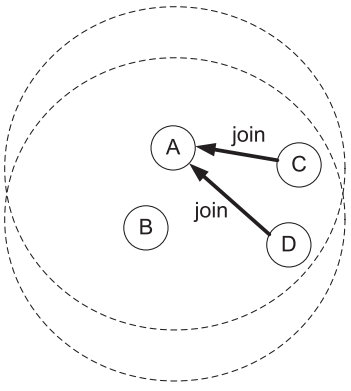


Fig. 3. One problem of the SWFS-U scheme.

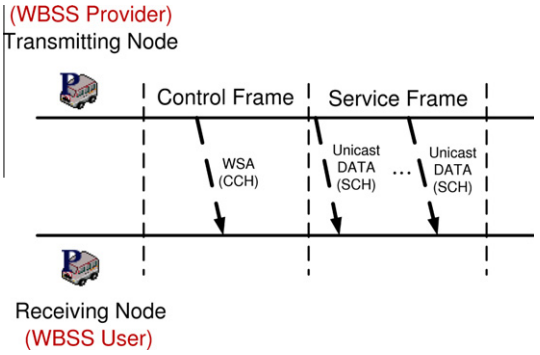


Fig. 4. The procedure to create a WBSS in the SWFS-U scheme.

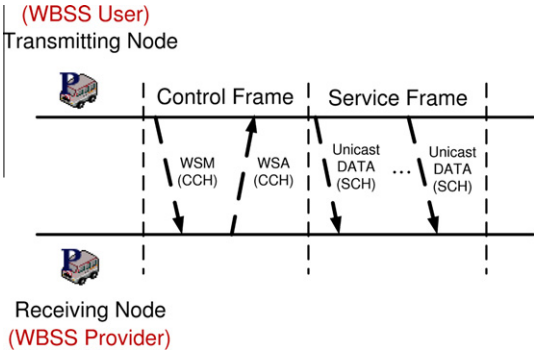


Fig. 5. The procedure to create a WBSS in the RWFS-U scheme.

that node *i* has data destined to node *k* and node *j* is the next-hop node for these data indicated by the upper-layer routing protocol. To send these data to node *j*, instead of broadcasting a WSA message on control frames on CCH, node *i* first broadcasts a "forward-req" WSM on control frames on CCH.

The "forward-req" WSM contains five fields: (1) the MAC address of the transmitting node; (2) the MAC address of the intended receiving node; (3) the transmission priority; (4) the sequence number of this WSM message; and (5) the ID of the SCH that the transmitting node intends to use. The first two fields are used to identify the transmitting and receiving nodes of the WSM. The transmission priority field is used by the receiving node to know whether it should service this request. The reason why a prioritized multihop forwarding design is required will be explained in Section 4.3. The sequence number field is used for the receiving node to know which WSM is the latest of all sent from a specific node. The sequence number is usually incremented by one at the beginning of each control frame and wrapped to zero when achieving the maximum value. The last field specifies which SCH is preferred by the transmitting node for data communication.

Suppose that node *i* just broadcast a "forward-req" WSM with the MAC address of the receiving node set to node *j*'s. Upon detecting the broadcast of a "forward-req" WSM, node *j* should receive this WSM. After receiving node *i*'s "forward-req" WSM, node *j* first compares the MAC address of the receiving node contained in the WSM with its

own MAC address. If the two MAC addresses differ, node *j* simply discards this WSM. Otherwise, node *j* then checks whether node *i* has the highest transmission priority among the transmitting node candidate list. The transmitting node candidate list should be maintained by each node using the information contained in received WSMs and WSAs. This list is composed of the nodes that had broadcast "forward-request" WSMs on the current control frames on CCH. (Note that this list includes node *j* itself, if it has data to send and has sent its "forward-req" WSM.)

If node *i* has the highest transmission priority, node *j* should create a WBSS for node *i* and broadcast a WSA message to notify it of the creation of this WBSS on control frames on CCH. Before transmitting the WSA out, node *j* should fill in the Provider Service Context (PSC) field of this WSA message with the MAC address of node *i* (i.e., the node that this WBSS intends to serve). According to the standard, the PSC field can be any arbitrary ASCII strings. Using this information, node *i* can know whether a WBSS has been created for it. The fact that the MAC address contained in the PSC field is the same as node *i*'s MAC address means that this WBSS is created for node *i*. Therefore, it joins this WBSS on the next service frame to transmit its data.

If node *j* receives a WSM from another node *k* with a transmission priority higher than node *i*'s and the MAC address of the receiving node carried in the WSM is its own MAC address, node *j* should service node *k*. This is accomplished by creating another WBSS for node *k* and broadcast a new WSA message for this new WBSS on control frames on CCH. Upon receiving node *j*'s later WSA for node *k*, node *i* will know that node *j* is forced to service another node with a higher transmission priority. Thus, it should stop the remaining data transmissions in the current transmission cycle, increase its transmission priority at the beginning of the next transmission cycle, and repeat the above process in the next transmission cycle. (Note that, our proposed protocol operates at the IEEE 1609.4 layer and does not control the operation of the IEEE 802.11(p) layer. Thus, node *i* will finish its current data transmission operation before stopping the remaining ones.)

Another case is that node *j* may receive a WSA broadcast by node *i*. This means that node *i* has abandoned its transmission request and intends to receive data from another higher-priority node. This is due to node *i*'s reception of another higher-priority node's WSM destined to it. In this case, node *j* can service another transmission request, if it exists. The WBSS that node *j* created for node *i* can be simply ignored because node *i* no longer uses it.

4.3. Proposed prioritized WBSS creation

Consider a scenario where two nodes intend to transmit data to each other in a WBSS-based vehicular network at the same time, and consequently, the two nodes simultaneously create their respective WBSSs. In this condition, neither of them will join the other's WBSS and complete their data transmissions. As a result, none of their data can be successfully sent and thus the end-to-end flow goodputs greatly decrease. To avoid this problem, we propose a prioritized WBSS creation mechanism to resolve WBSS creation conflicts and make nodes create WBSSs with fair chances. The proposed prioritized WBSS creation mechanism can collaborate with both of our proposed SWFS-U and RWFS-U schemes. Its details are explained below.

Initially, each 802.11(p) node sets its own transmission priority (for the multihop forwarding service) to 1. Consider that node *i* has data to send. As explained previously, it should form a WBSS and broadcast a WSA message on control frames on CCH. If node *i* succeeds in finding its intended receiving node *j* and transmitting unicast packet to node *j* on the next service frame on SCH, it should decrease its priority to zero (the lowest priority) in the next transmission cycle (i.e., at the beginning of the next control frame). Setting the transmission priority to the lowest value means that node *i* intends to suspend its WBSS creation, if there are other nodes intending to transmit data.

The success of unicast packet transmissions can be detected by the reception of the corresponding 802.11 ACK frames for the transmitted unicast packets. If node *i* fails in transmitting its unicast data, it should increase its priority by one in the next transmission cycle. For node *i*, the cases of failures of unicast data transmission are not the same in SWFS-U and RWFS-U. The common case of failing in unicast data transmissions in SWFS-U and RWFS-U is that node *i* has sent its unicast data out but not received any corresponding ACK frames on the same service frame.

In SWFS-U, failed unicast data transmissions can also be attributed to the following case: node *i* abandoned its created WBSS due to the reception of a higher-priority WSA message broadcast by its neighboring node (e.g., node *k*). This WSA message indicates that node *k* is going to start unicast data transmissions with a priority higher than node *i*. In this condition, under SWFS-U node *i* should abandon its own WBSS and join the WBSS created by node *k*. Similarly, in RWFS-U a node may fail in unicast data transmissions due to the reception of WSM or WSA messages from another neighboring node with a higher priority and destined to it. In this case, the node is forced to serve the data transmissions of another higher-priority node.

If the priority of a node is zero, the node is allowed to increase its priority by one on the next control frame. By this way, a transmitting node that has just successfully transmitted unicast data on the previous transmission cycle will yield the contention for creating a WBSS for one transmission cycle, letting its neighboring nodes have chances to initiate their data transmissions.

The performance gains of SWFS-U and RWFS-U due to the proposed prioritized WBSS creation mechanisms are presented in Section 5. Figs. 6 and 7 show the detailed state diagrams of SWFS-U and RWFS-U with the proposed prioritized WBSS creation mechanism, respectively. In these two state diagrams, an arrow denotes a state transition and is represented in the form of *A*/*B*, where *A* denotes the conditions that trigger this state transition and *B* denotes the actions that the scheme will take upon this state transition.

The "my" structure stores the necessary information for creating or joining a WBSS on the local node. It includes three fields: (1) priority (denoted as "pri" in the figures

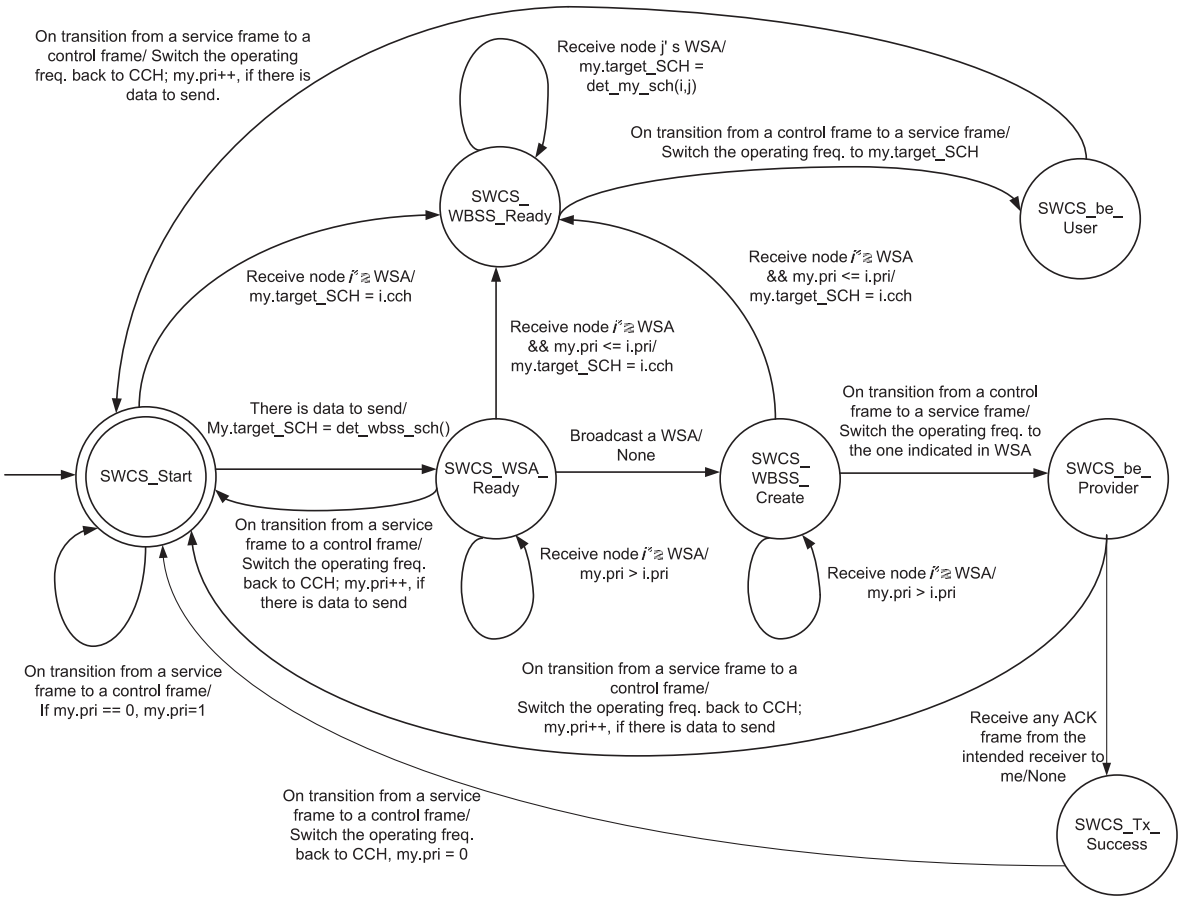


Fig. 6. The state diagram of SWFS-U.

for brevity); (2) target_sender; and (3) target_SCH. The priority field stores the transmission priority of the local node; the target_sender field stores the sender node chosen by the local node to serve (only valid in RWFS-U); and the target_SCH field specifies which SCH will be used by the local node to create/join a WBSS (if needed). The det_my_sender() and det_my_SCH() functions are used to determine the sending node that the local node should serve (only used in RWFS-U) and the SCH that the local node should switch into on the next service frame. The high-level operations of these two functions have been explained in previous sections. The details are not shown here for brevity.

4.4. SCH selection

The IEEE 1609.3 specification [8] suggests that each node chooses the least used SCH to create its own WBSS. We implemented this channel selection design in the proposed SWFS-U and RWFS-U schemes. In SWFS-U, each node maintains a channel utilization table based on its received WSA messages on control frames. The channel utilization table is used to record the number of active WBSSs on each SCH. With this information, each node can choose the least-used SCH to create its own WBSS when it needs to transmit data. This way, traffic of different nodes can be spread across all SCHs to efficiently utilize network bandwidth.

Such a channel selection design implemented in RWFS-U is similar to that implemented in SWFS-U, except that in RWFS-U each node should maintain its channel utilization table based on both of its received WSA messages and "forward-req" WSMs. The reason is that in RWFS-U the "forward-req" WSMs also carry the channel selection information of neighboring nodes. In a multihop and distributed environment, such an SCH selection design may not achieve the optimal network performances due to the hidden terminal problem. In Section 5, we study the performances of the SWFS-U and RWFS-U schemes with this SCH selection design and compare them with those under the optimal SCH selection condition.

In this paper, three SCH selection designs are evaluated. The first design is to choose an SCH using the hash-based formula ($\$nid \bmod N_s$), where $\$nid$ denotes a unique non-negative integer assigned to each node (starting from 1) and N_s denotes the number of SCHs in the network (which is six in the IEEE 802.11(p)/1609 network). Because the number of nodes in our first two example topologies does not exceed N_s (and thus each of them uses a different SCH), this hash-based design can generate the optimal SCH

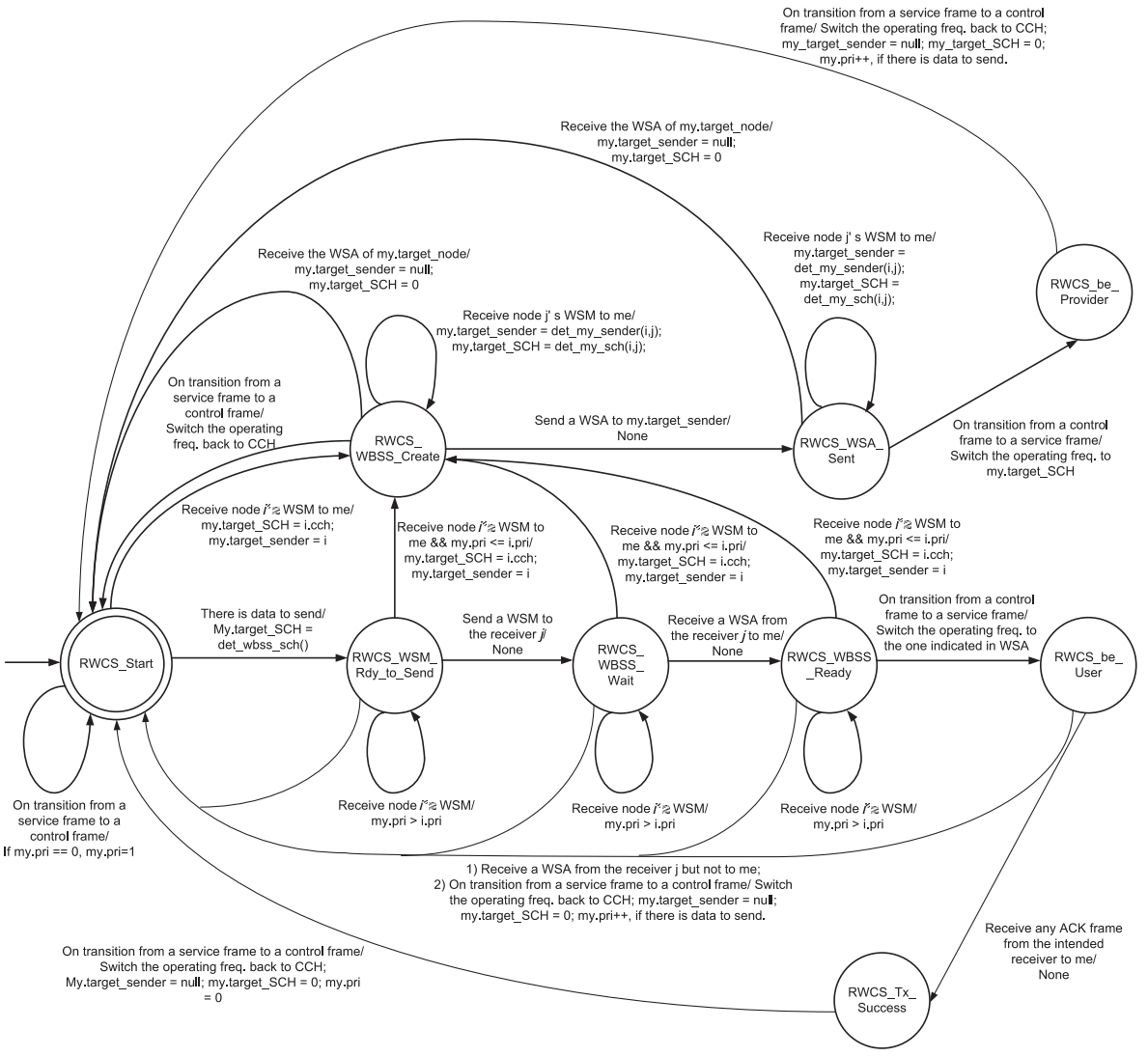


Fig. 7. The state diagram of RWFS-U.

assignment in our evaluated topologies. Thus, we call the SWFS-U and RWFS-U schemes using this hash-based SCH selection design SWFS-U (OSS) and RWFS-U (OSS), respectively.

The second design is to choose an SCH in a random manner. That is, when creating a WBSS, each node randomly choose an SCH among all SCHs. We denote the SWFS-U and RWFS-U schemes using the random SCH selection design as SWFS-U (RSS) and RWFS-U (RSS), respectively. The third design is to choose an SCH using the least used SCH design suggested by the standard. The SWFS-U and RWFS-U schemes using this channel selection design are denoted as SWFS-U (LSS) and RWFS-U (LSS), respectively.

5. Performance evaluation

In this section, we first build an analytical model to characterize the end-to-end goodputs of a unicast flow in IEEE 802.11(p)/1609 networks and then conduct simulations to extensively study the performances of SWFS-U and RWFS-U in detail.

5.1. Theoretical end-to-end flow goodputs

Given a network $N = (V, C)$, where V is the set of nodes in the network. $C = \bigcup_{i,j \geqslant 1, i \neq j}^{|V|} C_{ij}$ is the set of channels between two nodes that can communicate with each other. For an IEEE 802.11(p)/1609 chain network, $C_{ij} = 6$, *if* $j = i + 1$ *or* $j = i - 1$, and $C_{ij} = 0$, otherwise. A channel in C_{ij} is denoted as $C_{ij}^k, 1 \leqslant k \leqslant 6$. The set of nodes on flow f's routing path is denoted as R_f. Let $x_{i,j}^f$ be a binary random variable (RV) indicating that, when node v_i intends to receive the unicast data of flow f (those data transmitted by the source node of flow f) on a given service frame j from its previous-hop node, whether the channel of the WBSS that node v_i intends to join (in SWFS-U) or create (in RWFS-U) conflicts

with those used by its neighboring interfering nodes (e.g., v_{i+1} in a chain network). $x_{i,j}^f$ can be formulated as follows:

$$x_{i,j}^f = \begin{cases} 1, & \text{if the channel conflicts will not occur on node } v_i \text{ when it intends to receive the data of flow } f \text{unicast(those transmitted by the source node on the service frame } j) \text{ from the previous-hop node,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Let $y_{i,j}^f$ be a binary indicator indicating that, when node v_i intends to forward the unicast data of flow f (those transmitted by the source node of f) on a given service frame j, whether the next-hop node of node v_i (e.g., v_{i+1} in a chain network) can receive the data. $y_{i,j}^f$ can be formulated as follows:

$$y_{i,j}^f = \begin{cases} 1, & \text{if next-hop node is available to receive node } v_i\text{'s unicastdata of flow } f \text{ transmitted by the source on service frame } j, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The unavailability of the intended receiving node n_r for node v_i is attributed to four reasons: (1) node n_r cannot successfully receive node v_i's forward-req WSM; (2) node v_i cannot successfully receive the WSA message transmitted by node n_r; (3) node n_r is going to serve another transmitting node with a transmission priority higher than v_i's; and (4) all of the SCHs on the network have been occupied by other nodes on next service frame. In the fourth case, without the use of the virtual carrier-sensing mechanism (such as the RTS/CTS design in IEEE 802.11 networks) hidden nodes will generate severe packet collisions on node n_r on the next service frame. In this condition, the effective end-to-end goodput of flow f on this service frame is zero, regardless of whether node n_r will serve node v_i on that service frame.

The lifetime of a WBSS is assumed to be the duration of a service frame, which is a common setting in IEEE 802.11(p)/1609 networks. With this assumption, if the channel conflicts occur on node v_i on a service frame, node v_i is assumed not to be able to receive any unicast data on the given service frame, because the previous-hop node and other nodes neighboring to node v_i will form hidden node pairs on that service frame.

Let S_j^f be the binary indicator variable denoting whether the unicast data transmitted by the source node of flow f on a given service frame j can successfully reach the destination node of flow f, which can be defined as follows:

$$S_j^f = \begin{cases} 1, & \text{if } x_{i,j}^f \text{ is 1 and } y_{i,j}^f \text{ is 1}, \quad \forall i \in R_f, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The expected end-to-end flow goodputs of flow f can be defined as follows:

$$G(f) = E\left(\frac{D_j^f}{T_j^f} \cdot S_j^f\right), \quad (4)$$

where D_j^f denotes the amount of data that can be transmitted out by the source node of flow f on the service frame j and T_j^f denotes the interval of the service frame j. Because the D_j^f and T_j^f are independent of S_j^f, $G(f)$ can be rewritten as follows:

$$G(f) = E\left(\frac{D_j^f}{T_j^f}\right) \cdot E\left(S_j^f\right). \quad (5)$$

In the networks where idle SCHs always exist for a receiving node, $x_{i,j}^f$ and $y_{i,j}^f$ are independent because the receiving node can always find an idle SCH to receive data (if it is smart enough to know which SCH is idle). For example, this condition can hold in a chain network where the SCH is chosen using the function ($nid mod N_s) discussed in Section 4.4. In this condition, $E(S_j^f)$ can be defined as $X_j^f Y_j^f$, where X_j^f denotes the expected value of the probability that channel conflicts do not occur on the route of flow f, when nodes along this route are forwarding data sent on the service frame j. Y_j^f denotes the expected value of the probability that the receiving nodes on the route of flow f are able to receive data from their previous-hop nodes. (The forwarded data are those transmitted by the source node of flow f on the service frame j.)

X_j^f and Y_j^f can be defined as follows, if $\langle x_{i,j}^f \rangle, \forall i$, is independent and identically distributed (i.i.d) and $\langle y_{i,j}^f \rangle, \forall i$, is i.i.d:

$$X_j^f = \sum_{k=0}^{n} C_k^n \left(x_{i,j}^f \cdot p\left(x_{i,j}^f\right)\right)^k \cdot \left(\overline{x_{i,j}^f} \cdot p\left(\overline{x_{i,j}^f}\right)\right)^{n-k}, \quad (6)$$

where $\overline{x_{i,j}^f}$ denotes $(1 - x_{i,j}^f)$.

$$Y_j^f = \sum_{m=0}^{n} C_m^n \left(y_{i,j}^f \cdot p\left(y_{i,j}^f\right)\right)^m \cdot \left(\overline{y_{i,j}^f} \cdot p\left(\overline{y_{i,j}^f}\right)\right)^{n-m}, \quad (7)$$

where $\overline{y_{i,j}^f}$ denotes $(1 - y_{i,j}^f)$. If $\langle x_{i,j}^f \rangle, \forall i$, and $\langle y_{i,j}^f \rangle, \forall i$, are not i.i.d, X_j^f and Y_j^f are defined as follows:

$$X_j^f = \overline{x_1^f} \cdot p\left(\overline{x_1^f}\right) + x_1^f \cdot p\left(x_1^f\right)\left\{\left(\overline{x_2^f} \cdot p\left(\overline{x_2^f}\right)\right. + \left(x_2^f \cdot p\left(x_2^f\right)\left[\ldots\left(\overline{x_n^f} \cdot p\left(\overline{x_n^f}\right) + x_n^f \cdot p\left(x_n^f\right)\right)\right]\right\}, \quad (8)$$

$$Y_j^f = \overline{y_1^f} \cdot p\left(\overline{y_1^f}\right) + y_1^f \cdot p\left(y_1^f\right)\left\{\left(\overline{y_2^f} \cdot p\left(\overline{y_2^f}\right)\right. + \left(y_2^f \cdot p\left(y_2^f\right)\left[\ldots\left(\overline{y_n^f} \cdot p\left(\overline{y_n^f}\right) + y_n^f \cdot p\left(y_n^f\right)\right)\right]\right\}. \quad (9)$$

Because $x_{i,j}^f$ and $y_{i,j}^f$ are binary indicators, regardless of whether they are i.i.d., X_j^f and Y_j^f can be reduced as follows:

$$X_j^f = \left(x_1^f \cdot p\left(x_1^f\right)\right)\left(x_2^f \cdot p\left(x_2^f\right)\right) \ldots \left(x_n^f \cdot p\left(x_n^f\right)\right), \quad (10)$$

$$Y_j^f = \left(y_1^f \cdot p\left(y_1^f\right)\right)\left(y_2^f \cdot p\left(y_2^f\right)\right) \ldots \left(y_n^f \cdot p\left(y_n^f\right)\right). \quad (11)$$

Thus, $G(f)$ can be reduced as follows:

$$G(f) = E\left(\frac{D_j^f}{T_j^f}\right) * \prod_{i=1}^{n}\left(p\left(x_{i,j}^f\right)\right)\prod_{i=1}^{n}\left(p\left(y_{i,j}^f\right)\right). \quad (12)$$

From Eq. (12), it is known that the end-to-end flow goodputs are greatly determined by the probabilities of the occurrences of channel conflicts and reception unavailability on the nodes along the flow's route. Denote $p_{c,j}^f$ as the probability that no channel conflicts occur on the nodes of flow f's route, when they are receiving the unicast

data of f transmitted by the source node on the service frame j from their previous-hop node (i.e., $\prod_{i=1}^{n} p\left(x_{i,j}^{f}\right)$), and $p_{s,j}^{f}$ as the probability that no receiving nodes are unavailable on flow f's route, when their previous-hop nodes are forwarding the unicast data of f transmitted by the source node on the service frame j to them (i.e., $\prod_{i=1}^{n} p\left(y_{i,j}^{f}\right)$) The designs of a multihop forwarding scheme (e.g., distinct WBSS creation schemes and SCH selection schemes) can greatly affect $p_{c,j}^{f}$ and $p_{s,j}^{f}$. In Section 5.2, we study the end-to-end flow goodputs that can be achieved by SWFS-U and RWFS-U and the relationship between the designs of multihop forwarding schemes and the values of $p_{c,j}^{f}$ and $p_{s,j}^{f}$.

5.2. Simulation results

The simulations were conducted using the NCTUns network simulator [12]. The main performance metric is the average end-to-end goodputs of all UDP flows, which is defined as $\frac{\sum_{i=1}^{N} G(i)}{N}$, where $G(i)$ is the average end-to-end goodput obtained by the ith greedy UDP flow and N denotes the total number of greedy UDP flows in a simulation run. A "greedy" UDP flow means that a UDP flow transmits data as much as possible each time when it is invoked by the operating system to transmit data. That is, a greedy UDP flow always exhausts the available bandwidth when transmitting data and is often used to estimate the maximum end-to-end throughputs that can be obtained by applications. The average end-to-end flow goodput of flow f (denoted as $G(f)$) is obtained as follows:

$$G(f) = \frac{\sum_{j=t_s}^{t_e} g_f(j)}{t_e - t_s}, \tag{13}$$

where t_s and t_e denote the time points where flow f starts and ends its data transmission, respectively. $g_f(j)$ denotes the number of bytes received at flow f's receiving program during the jth second. Each result presented in this paper is the average across ten simulation runs, each using a different random number seed. The transmission range and the interference range of each IEEE 802.11(p) radio is set to 720 m and 884 m, respectively. The channel model used in the simulations is the two-ray model. The intervals of a control frame and a service frame are set to 50 ms. The modulation and coding rate used in the simulations are QPSK and 1/2, respectively. With these settings, the MAC-layer data rate is 6 Mbps and thus the data rate on a service frame on an SCH is 3 Mbps. Each node is associated with a unique $nid (starting from 1) in simulations. Note that each point of the average end-to-end goodput results shown in the following sections is plotted with its standard deviation. However, these standard deviation values are usually very small (e.g., below 1 KB/s). Thus, it is insignificant to the average value and difficult to be seen in the figures.

5.2.1. Chain network topology

We first study the performances of SWFS-U and RWFS-U in chain network topologies, which is common on highways or on the roads in rural areas. Nine chain topologies are built to create N-hop chain network scenarios, where $1 \leqslant N \leqslant 9$. For example, an N-hop network is composed of $N+1$ nodes, each is spaced 500 m away. In each topology, a flow with the left-most node as the source node and the right-most node as its destination node generates greedy UDP traffic. The end-to-end flow goodput results across different hop counts are plotted in Fig. 8.

We first compare the end-to-end goodputs of SWFS-U (OSS) and RWFS-U (OSS). With the OSS design, $p_{c,j}^{f}$ is always 1 for each WBSS in chain networks. In this condition, the drop of end-to-end goodputs is mainly attributed to the unavailability of receiving nodes, which results from the half-duplex nature of the wireless medium: an 802.11 radio cannot simultaneously send and receive data. That is why a drastic 50% decrease of the flow goodputs occurs from 1 hop to 2 hops. In a multi-channel chain network, if $p_{c,j}^{f}$ is 1, which means that clear channels always exist, the $p_{s,j}^{f}$ value of a route can optimally remain at 0.5. This means that the forwarding efficiency (defined as the end-to-end flow goodputs divided by the data generation rate of the source node) of a multihop path can optimally be 50%. One can see that RWFS-U can achieve this optimal performance in a chain network with the optimal SCH arrangement.

In contrast, due to the sender-centric design, the forwarding efficiency of SWFS-U significantly decreases when the hop count of a route increases. Denote $p_{s,j}^{f,k}$ as the $p_{s,j}^{f}$ value for a k-hop chain network. $p_{s,j}^{f,2} = 0.5$ represents the optimal availability probability of a receiving node for a multihop flow in chain networks. As shown in Fig. 9, nodes using RWFS-U can remain the optimal service availabilities for multihop traffic in a multi-channel chain network, regardless of the hop count of the route, while nodes using SWFS-U significantly decreases their service availabilities as the hop count of the route increases.

We then compare the goodput performances of RWFS-U (OSS) and RWFS-U (LSS). Because each node cannot obtain the SCH information of all its two-hop neighboring nodes, the LSS design cannot achieve the optimal SCH arrangement. A receiving node j using RWFS-U can passively avoid packet collisions due to channel conflicts. This is accomplished by simply ignoring the "forward-req" WSM of node i that intends to use an SCH that will generate a channel

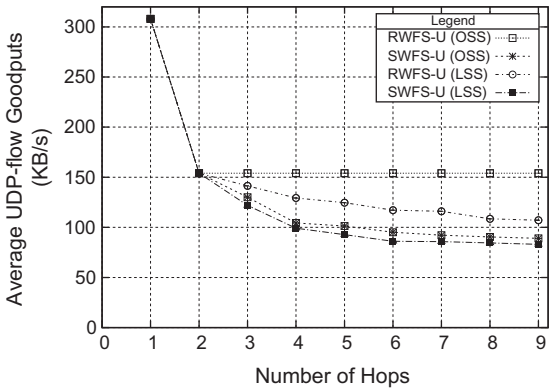


Fig. 8. The average end-to-end goodputs in chain networks.

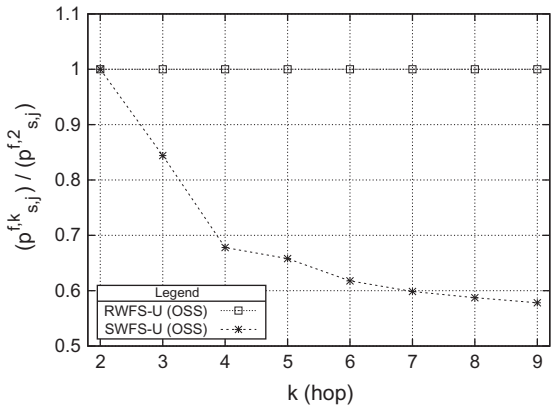


Fig. 9. The ratio of $p^{f,k}_{s,j}$ to $p^{f,2}_{s,j}$ across different hop counts.

conflict. This approach, however, may waste the bandwidth of a service frame on nodes *i* and *j*.

This means that the passive design used in RWFS-U is insufficient to achieve the optimal multihop forwarding performances and a design that can actively propagate the information of SCHs used by each WBSS to two-hop neighboring nodes is required to do so. However, the current formats of the control messages defined by the WAVE mode do not carry such information and need to be expanded to achieve this goal. An example proposal to this enhancement is [18], where the authors modified the control messages of the WAVE mode to actively alert channel conflicts between neighboring WBSSs and propagate the SCH information of each WBSS. The modifications to the 802.11(p)/1609 control messages are out of the scope of this paper and not discussed here.

We finally show the effects of the prioritized WBSS creation design on end-to-end multihop flow goodputs in chain networks. As shown in Fig. 10, our proposed prioritized WBSS creation design can significantly increase the end-to-end flow goodputs of RWFS-U and SWFS-U. For example, using this design the 4-hop flow goodputs of RWFS-U can be increased by 35.82% and the 5-hop flow goodputs of SWFS-U can be increased 26.68%, respectively.

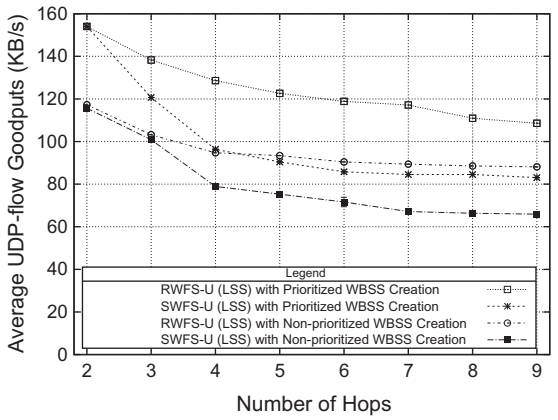


Fig. 10. The effects of prioritized WBSS creation on RWFS-U and SWFS-U.

The main reason is that prioritized WBSS creation guarantees that, after successfully transmitting data, a node will yield creating a new WBSS for one transmission cycle, if other neighboring nodes intend to transmit data (i.e., create their own WBSSs). For a multihop flow, if an intermediate forwarding node receives excessive data but has no chances to send them out, these received data will be dropped at the packet output buffer due to the lack of output buffer space. Because the prioritized WBSS creation design can help neighboring transmitting/forwarding nodes fairly create their respective WBSSs on each transmission cycle, the occurrences of packet drops on forwarding nodes can be reduced. Thus, it can increase the end-to-end goodputs of multihop flows. In this paper, if not explicitly indicated, the performance results of RWFS-U and SWFS-U are presented with the presence of the proposed prioritized WBSS creation design.

5.2.2. Dense network topology

We then study the end-to-end flow goodputs under SWFS-U and RWFS-U in a dense network topology, where 12 nodes (OBUs) are deployed on a two-lane road segment with the width and the length set to 15 m and 100 m, respectively. In this network topology, all nodes can listen to and interfere with each other. This scenario is likely to happen at the intersection area where several vehicles may stop moving due to the stop sign of the traffic light at the intersection.

Six flows are created to generate greedy UDP traffic in this network. Each flow i, $1 \leqslant i \leqslant 6$, uses node i as its source node and node $(i+6)$ as its destination node. Each node is randomly placed within the road segment in each simulation run. We varied the number of activated flows and plotted the average end-to-end flow goodput results obtained using simulations in Fig. 11.

Each source node can directly communicate with its destination node; thus, $p^{f}_{s,j}$ is 1 for all traffic flows using RWFS-U due to the polling-response design of RWFS-U. In contrast, traffic flows using SWFS-U cannot achieve the 1.0 value of $p^{f}_{s,j}$ because, when receiving a WSA message from node i, all other nodes that receive this WSA message should first join the WBSS created by node i to

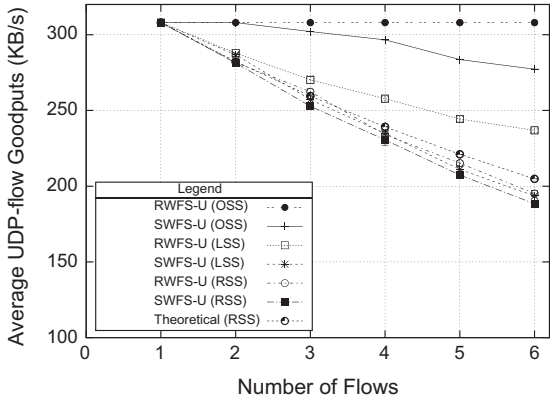


Fig. 11. The average flow throughputs in the tested dense network topology.

check whether there are any data for them. This try-and-error design can make nodes unnecessarily join a WBSS, decreasing the service available time of forwarding nodes for multihop flows.

This phenomenon can be evidenced by comparing the end-to-end flow goodputs of RWFS-U (OSS) and SWFS-U (OSS). The $p^f_{c,j}$ values of these two schemes are 1 in this network case due to the optimal SCH selection. Because RWFS-U can achieve the 1.0 value of $p^f_{s,j}$ with the polling-response model, it can generate the optimal end-to-end flow goodputs over all of the tested hop counts in this network. In contrast, the end-to-end goodputs of a flow using SWFS-U decrease when the number of active flows in a collision domain increases, due to the decrease of receiving nodes' service availability. For example, as shown in Fig. 11, $p^f_{s,j}$ of SWFS-U under the 5-flow scenario is 90.03% only. The $p^f_{s,j}$ results of RWFS-U and SWFS-U show that a receiver-centric design is more suitable for transmitting multihop and heavy-load traffic in WBSS-based networks.

The performance of the SCH selection designs are discussed here. Given a dense all-connected 1-hop network $N = (V,C)$, $|V| = n$. The average flow goodputs in N can be modeled as follows:

$$G(f) = \sum_{i=1}^{n} (|G_i|) \cdot \left(P^i_{ch}\right), \tag{14}$$

where G_i denotes the throughputs obtained by a flow, when i flows fairly share the same SCH. P^i_{ch} denotes the probability that i flows use the same SCHs to transmit data. Because all nodes can interfere with and sense each other in this network topology, no hidden node pairs occur. Thus, G_i can be $B_{ch} * \frac{1}{i}$, where B_{ch} is the bandwidth of an SCH.

Suppose that the first flow has chosen an SCH for data transmission. Let p_i be the probability that the remaining flows choose the same SCH chosen by the first flow to transmit data. $G(f)$ can be represented as follows:

$$G(f) = \sum_{i=1}^{n} \frac{B_{ch}}{i} \cdot C^{n-1}_{i-1} \cdot (p_i)^{i-1}(1-p_i)^{n-i}. \tag{15}$$

The OSS design can achieve the optimal SCH assignment in this dense network case. This means that p_i is 0 and the resulting $G(f)$ is B_{ch} (matching the simulation results shown in Fig. 11). However, the OSS design can achieve the optimal SCH selection only when (1) the number of transmission-reception node pairs is less than the number of idle SCHs in the network and (2) neighboring nodes are assumed not to generate the collisions of hash values (which are used to determine the SCHs used by nodes).

These two conditions may not be met in real vehicular networks, which means that a practical SCH selection design is desired in real WBSS-based networks. We now evaluate the performances of the LSS and RSS designs (explained in Section 4.4), which are two feasible designs in the current 802.11(p)/1609 standards.

For the RSS design, p_i is $\frac{1}{|C_{ij}|}$, where $|C_{ij}|$, $1 \leqslant i, j \leqslant n$, is the number of useable SCHs (which is six) in the standards. Thus, $G(f)$ can defined as follows:

$$G(f) = \sum_{i=1}^{n} \frac{B_{ch}}{i} \cdot C^{n-1}_{i-1} \left(\frac{5^{n-i}}{6^{n-1}}\right). \tag{16}$$

The average flow throughputs derived by Eq. (16) is denoted as theoretical (RSS) in Fig. 11, which greatly matches those of RWFS-U (RSS) obtained in the simulation experiments.

Notice that the LSS design is supposed to achieve the optimal end-to-end flow goodputs in this dense 1-hop connected network. However, the simulation results of the end-to-end flow goodputs using the LSS design in this network topology are not optimal. The reason is that, the WME of a node creates a WBSS with a chosen SCH each time when upper-layer applications push data to it and its own WBSS does not exist. When a WBSS is created, it will not change its SCH for one transmission cycle. Thus, although the used SCH information is updated after receiving "forward-req" WSMs and WSAs from neighboring nodes, the WBSS that has been created will not change its used SCH in this transmission cycle.

The implementation of the LSS design can be improved using a cross-layer design. For example, allowing the WME to change the SCH of a WBSS when the "forward-req" WSM of this WBSS is generated and queued at the output packet queue (i.e., it has not been sent out). This cross-layer design can a bit reduce the number of channel conflicts by changing the SCH of a WBSS before it is announced, at a cost of increased implementation complexity. Another possible design is to allow the WME to change the SCH of a WBSS on the same control frame. Such designs need more heuristics to avoid the oscillation of SCH selections among neighboring WBSSs (especially in mobile vehicular networks), which may cause nodes to excessively generate control messages on control frames for unnecessary SCH re-selection.

5.2.3. Grid network topology

The performance of RWFS-U and SWFS-U in multi-flow multihop scenarios are studied here. We created five grid network topologies, each composed of 4 (2×2), 9 (3×3), 16 (4×4), 25 (5×5), 36 (6×6) nodes, respectively. Each node is spaced 700 m away in the simulated networks and can communicate with only its vertical and horizontal neighboring nodes. For clarity, each node of the created grid network is assigned a unique ID, increased by one, from the left to the right, and from the top to the bottom. For example, the IDs of the first row of the 6x6 grid network are from 1 to 6 and those of the second row of the network are from 7 to 12.

Each $N \times N$ grid network has a greedy UDP flow on each of its rows. For the flow on a row, the left-most node on the row is its source node and the right-most node is its destination node. To avoid channel conflicts between neighboring WBSSs, nodes using the OSS design choose the SCH used by their WBSSs with Eq. (17).

$$sch\_id = \begin{cases} (nid-1) \bmod N_s, & \text{if } ((nid-1)/N_s \bmod 2) = 0, \\ (nid-1+3) \bmod N_s, & \text{otherwise}, \end{cases} \tag{17}$$

where N_s denotes the number of SCHs. Fig. 12 shows the average end-to-end flow goodputs in the simulated grid

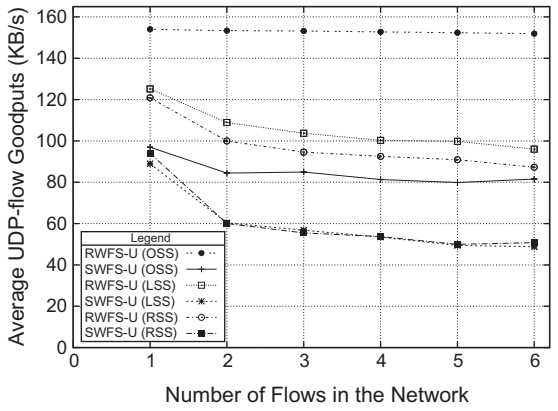


Fig. 12. The average flow goodputs in grid networks.

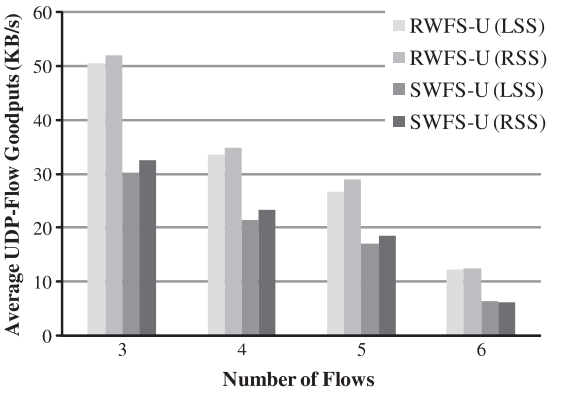


Fig. 13. The average flow goodputs in the 6×6 grid network with diagonal traffic patterns.

networks. RWFS-U (OSS) on average can outperform SWFS-U (OSS) on end-to-end flow goodputs by factors of more than 1.80, when multiple multihop flows exist. Because these two schemes use the optimal channel assignment, which eliminates channel conflicts in the tested grid networks, the difference of flow goodput results between them shows how well these two schemes arrange concurrent data transmissions in the tested networks.

Under SWFS-U, the only way for a node to know whether there are any data destined to it in a WBSS is to join that WBSS and check the destination address of packets received in that WBSS. With this constraint, suppose that node *i* just broadcasts the WSA of its WBSS which has the highest transmission priority in its two-hop neighborhood. All of its one-hop neighboring nodes will join node *i*'s WBSS on the next service frame. In this condition, nodes that are not node *i*'s intended receiver may waste the bandwidth of the service frame, because if they have data to send, they need to create their own WBSSs with their respective intended receivers. However, this cannot be done until the next control frame. Formally, under SWFS-U node i makes $|\mathrm{nbr}_1(i)|$ nodes join its WBSS and blocks $|\mathrm{nbr}_1(i)|$ data transmissions in its neighborhood, where $\mathrm{nbr}_1(i)$ is the number of one-hop neighboring nodes of node i.

In contrast, under RWFS-U a transmitting node can effectively create its WBSS exactly (and only) with its intended receiver. Thus, other neighboring nodes can create their WBSSs at other SCHs to concurrently transmit data on the same service frame. As a result, nodes using RWFS-U can more effectively use network bandwidth when multiple co-interfering flows exist, as compared with those using SWFS-U.

The average end-to-end flow goodputs of nodes using the LSS and RSS designs are approximately 60% only of the end-to-end flow goodputs of nodes using the OSS design, when multiple co-interfering flows exist. The reason is that the LSS and RSS designs cannot effectively eliminate channel conflicts in such scenarios, which frequently take place in real vehicular networks. The flow goodput results of the LSS design (which is suggested in the 802.11(p)/1609 standard family) are only a bit better than those of the RSS design under RWFS-U and degenerate to those of the RSS design under SWFS-U, which shows that, in addition to the WBSS creation scheme, the design of selecting SCHs for created WBSSs is also significant to the end-to-end flow performances.

In the evaluated grid network, the maximum number of co-interfering flows (including the flow itself) is 3, when the number of flows exceeds 3. Thus, the end-to-end flow goodputs of nodes using the LSS and RSS designs remain nearly the same when the number of co-interfering flows is larger than 3.

We finally study the flow goodput performance of the RWFS-U and SWFS-U under more complicated traffic patterns in the created 6×6 simulated grid network. Two diagonal greedy UDP flows are added as the background traffic. A number of greedy UDP flows are added in sequence to increase the traffic load in the network.[1] The results are plotted in Fig. 13. The standard deviation of each point is below 1 KB/s and thus is not shown in Fig. 13. Note that, the flow goodput results of the OSS design are not shown in Fig. 13 because, under the cross-row traffic patterns, the OSS design cannot generate an optimal SCH assignment to eliminate channel conflicts. In this condition, the performance results of the OSS design would be misleading.

For the results of Fig. 13, two findings are worth noticing. First, under the heavy traffic loads RWFS-U significantly outperforms SWFS-U on end-to-end flow goodputs. This evidences that RWFS-U can effectively increase $p^f_{s,j}$ by using its per-hop two-way handshake design. Second, comparing the goodput results of the LSS design and those of the RSS design, one can find that, when most of nodes are actively transmitting and receiving data with multiple neighboring nodes, the LSS design performs the same as or a bit worse than the RSS design on the goodput results. The reason is that, when most of nodes are transmitting or receiving data at the same time, channel conflicts are inevitable if each node selects the used SCH only using its own local knowledge. In this condition, exploiting randomness is a simple yet effective method to select an SCH.

[1] The added UDP flows are listed as follows in the order as they are added into the network with the (source node, destination node) representation: (13,24), (3,34), (18,19), (33,4).

6. Related work

To reduce packet collisions and transmission latencies in highly dynamic CSMA/CA-based vehicular networks, slot-based channel access schemes were proposed in [13,14] and TDMA-based schemes were proposed in [15,16]. The objectives of these previous papers greatly differ from those of ours. These previous papers focus on the performances of 1-hop data communications in 802.11(p) networks, while our paper studies the impacts of the two WBSS-creating schemes on multihop unicast traffic forwarding in WBSS-based 802.11(p)/1609 networks.

In [17], the authors observed that the users of a WBSS are continuously changing over time due to the high-mobility property of vehicular networks. In this condition, a WBSS provider may unnecessarily transmit and retransmit data packets to its WBSS users that have moved out of its signal coverage, wasting link bandwidth. To address this problem, they proposed a new WBSS-user-oriented WAVE mode for IEEE 802.11(p)/1609 networks. Using this new operational mode, a WBSS provider will transmit data packets to its users only when it receives a "transmission request" control message sent by them.

The WBSS-creating scheme proposed in [17] falls into the sender-centric design category as well because a WBSS is created by the transmitting node. Although this user-initiated polling design can save link bandwidth by reducing unnecessary packet transmissions, it requires each node to periodically join and poll all neighboring WBSSs to know whether there is data destined to it. Compared with our proposed RWFS-U, such a polling mechanism is more time-consuming to forward packets among nodes.

In [18], the authors proposed a gossiping-based scheme for WBSS providers to exchange the SCHs that they choose. A "channel assignment warning" message can be issued by nodes neighboring to these WBSS providers, if they found that multiple neighboring WBSS providers choose the same SCH. This scheme, however, needs to add a new control message in the network and modify the format of the WSA message. As a result, it cannot operate on the current IEEE 802.11(p) network but can be considered as a possible extension for future vehicular network.

In [19], the authors conducted simulation studies on the performances of multihop forwarding using several unicast routing protocols (e.g., AODV and DSDV) in vehicular networks. The main performance parameter is the average of packet delivery ratios of end-to-end flows. Although the authors presented some initial simulation results on the performances of multihop forwarding in vehicular networks, some of the important information about their simulation experiments is unclear. For example, it is unclear that whether the authors used a complete 802.11(p)/1609 MAC implementation to conduct simulations and how the unicast routing protocols used in this work disseminate their control messages in WBSS-based networks. In our work, we used the complete 802.11(p)/1609 implementation provided by NCTUns to conduct simulations, discussed and solved the issues of deploying unicast routing protocols in WBSS-based networks. In addition, we proposed a receiver-centric WBSS creating scheme to enhance the multihop forwarding performances for unicast traffic in WBSS-based networks.

7. Conclusion

In this paper, we proposed a Receiver-centric Multihop Forwarding Scheme for Unicast data (denoted as RWFS-U) for forwarding unicast data in WBSS-based vehicular networks. RWFS-U uses a new receiver-centric WBSS-creating scheme to create WBSSs. In addition, it employs a prioritized WBSS to further improve the forwarding performances in multihop scenarios.

The performances of RWFS-U and traditional Sender-centric Multihop Forwarding Scheme for Unicast Data (denoted as SWFS-U) are evaluated using both analytical and simulation approaches. The results show that RWFS-U significantly outperforms SWFS-U on end-to-end flow goodputs in most of the studied network scenarios. Given the restrictions imposed by the current IEEE 802.11(p)/1609 standards on a WBSS-based network, our proposed RWFS-U is quite efficient in serving multihop unicast traffic in WBSS-based 802.11(p)/1609 networks.

Appendix A

In Appendix A, we evaluated the impacts of the ways to disseminate the control messages of routing protocols on their routing performances. The simulations are conducted using the chain network scenario in Section 5.2.1. Two classic routing protocols are evaluated: DSDV [10] and AODV [11]. Two performance metrics are studied. One is the path establishment time of a route, which is defined as the average time required by a routing protocol to detect the route for a flow, after the network boots. The other metric is the path recovery time of a route, which is defined as the average time required by a routing protocol to detect the breakage of a route and recover it using another possible path. The performance results of these two routing protocols are presented below.

DSDV is a proactive routing protocol that periodically broadcasts its local information to neighboring nodes. Its details can be found in [10]. In our implementation, the broadcast is accomplished by using WSMs, which can be broadcast on control frames (and possibly on service frames) in WBSS-based networks. Fig. A.14 shows the path establishment times of the DSDV protocol in legacy 802.11(a) chain networks and 802.11(p)/1609 chain networks. The results are expected. By using WSMs to broadcast control messages, DSDV in the 802.11(p)/1609 network can perform as well as that in the 802.11(a) network.

In contrast, AODV is a reactive routing protocol that finds a route only when it is needed. Using AODV, when probing a route towards a certain destination node, the source node first broadcasts a "Route Request (RREQ)" messages towards to the destination node. Intermediate nodes that receive such a RREQ message will re-flood it towards the destination node. At the same time, the intermediate nodes establish a route backward to the source node

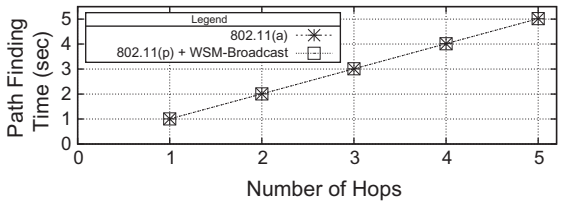


Fig. A.14. The path establishment time of the DSDV protocol in chain networks.

based on the transmitting node of the received RREQ message. The flooding process ends when the RREQ message reaches the destination node.

The destination node then transmits a "Route Reply (RREP)" message back to the source node using unicasts. Intermediate nodes can know how to route the RREP message back to the source node by using the previously established backward routes to the source node. The path establishment time for AODV can be defined as the interval between the time point that the source node broadcasts the RREQ messages and the time point that it receives the RREP message of the destination node. Recall that in our implementation the broadcast of RREQ messages is accomplished by using broadcast WSMs. Unicast RREP messages can be transmitted using either IP packets or WSMs. We evaluated the path establishment times of the AODV protocol with RREPs transmitted using these two approaches below.

The path establishment time results of AODV are plotted the results in Fig. A.15. Because IP packets can be transmitted using RWFS-U and SWFS-U, the path establishment times of AODV under both schemes are separately evaluated. As shown in Fig. A.15, the path establishment time of AODV increases as the hop count of the route increases, when RREP messages are transmitted using IP packets regardless of the used forwarding schemes. The reason is that, when transmitting RREP messages using IP packets, each node should first create its own WBSS on a control frame and then transmit RREP messages on next service frame.

Consequently, each RREP forwarding requires a transmission cycle (which is 0.1 s in our simulations). For an N-hop path, the path establishment time of AODV with RREPs transmitted using IP packets can be modeled as follows:

$$\text{Path establishment time} = N_h \cdot (Tx_{rreq} + Tp_{rreq}) + N_h \cdot (T_{cf} + T_{sf}) + T_{ex}, \quad \text{(A.1)}$$

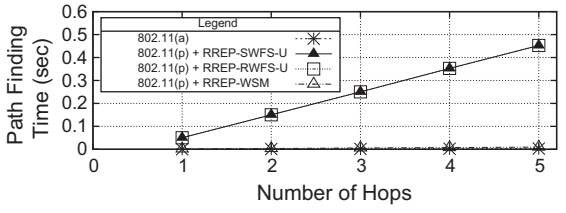


Fig. A.15. The path establishment time of the AODV protocol in chain networks.

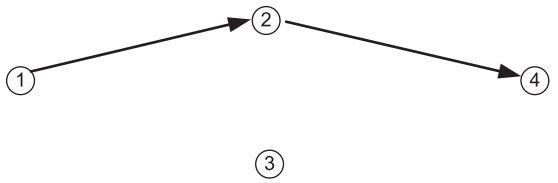


Fig. A.16. Two-hop link breakage scenario.

Table 1
The path recovery times of the DSDV and AODV protocols in chain networks.

Routing protocol/network	802.11(a)	802.11(p) with WSM
DSDV	1037 ms	1033 ms
AODV (RREP-WSM)	191 ms	214 ms

where N_h denotes the hop count of the route. Tx_{rreq} denotes the transmission time required by the 802.11(p) radio to transmit a RREQ message out and Tx_{rrep} denotes the propagation delay for the RREQ message to reach the next-hop node. T_{cf} and T_{sf} denote the intervals of a control frame and a service frame, respectively. T_{ex} represents the potential waiting time overhead if (1) the medium is too busy to transmit the RREQ message out within one control frame (due to an excessive number of contending nodes) or (2) the elapse time of the current control frame is not long enough to transmit it out (depending on when the node transmits the RREQ out). In our evaluated chain topology, T_{rreq}, T_{rrep}, and T_{ex} are tiny as compared with $(T_{cf} + T_{sf})$. The path establishment time of AODV with RREP transmitted using IP packets is therefore linear to $(T_{cf} + T_{sf})$.

We then study the path recover times of the DSDV and AODV protocols using a two-hop network topology shown in Fig. A.16, where each node is only able to communicate with its neighboring nodes. Suppose that node 1 intends to transmit data to node 4. Initially, node 3 turns off its radio. Thus, node 1 chooses node 2 as the forwarding node. At the 10th second, node 2's radio will be turned off and node 3 will turn on its radio. As a result, the routing protocols on node 1 will soon detect the route failure and eventually finds the new route that passes through node 3 to node 4.

Table 1 shows the path recovery times of the DSDV protocol with broadcast control messages transmitted using broadcast WSMs and the AODV protocol with RREPs transmitted using WSMs in this example scenario. The key point is that, when using WSMs as the facility to broadcast/transmit control messages, the DSDV and AODV protocols WBSS-based 802.11(p)/1609 networks can perform as well as those in the traditional 802.11(a) network. The path recovery time of AODV in 802.11(p) network is slightly higher than that in 802.11(a) network. The reason is that the link failure may occur on service frames; however, nodes in the 802.11(p) network are only allowed to initiate the route recovery process on control frames. In this condition, the transmissions of such route recovery messages will be postponed until next control frame arrives, which slightly increases the delay for discovering a new route. (Note that the "Route Error (RERR)" message is either transmitted or broadcast using WSMs, if broadcast is more appropriate and more efficient.)

References

[1] S.Y. Wang, C.C. Lin, K.C. Liu, W.J. Hong, On multihop forwarding over WBSS-based IEEE 802.11(p)/1609 networks, in: The 20th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2009), September 13–16, 2009, Tokyo, Japan.

[2] P802.11p/D9.0, Draft Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks –Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 7: Wireless Access in Vehicular Environments, July 2009.

[3] IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), June 12, 2007.

[4] IEEE P1609.0/D0.7 Draft Standard for Wireless Access in Vehicular Environments (WAVE) – Architecture, the WAVE Working Group of the ITS Committee, January, 2009.

[5] S.Y. Wang et al., Evaluating and Improving the TCP/UDP Performances of IEEE 802.11(p)/1609 Networks, in: Proceedings of the IEEE Symposium on Computers and Communications 2008 (ISCC 2008), July 6–9, Marrakech, Morocco, 2008, pp. 163–168.

[6] IEEE 1609.1 Trial-Use Standard for Wireless Accesses in Vehicular Environments (WAVE) – Resource Manager, IEEE Vehicular Technology Society, October 2007.

[7] IEEE 1609.2 Trial-Use Standard for Wireless Accesses in Vehicular Environments (WAVE) – Security Services for Applications and Management Messages, IEEE Vehicular Technology Society, October 2006.

[8] IEEE 1609.3/D1.0 Draft Standard for Wireless Accesses in Vehicular Environments (WAVE) – Networking Services, the WAVE Working Group of the ITS Committee, December 2008.

[9] IEEE 1609.4/D1.0 Draft Standard for Wireless Accesses in Vehicular Environments (WAVE) – Multi-channel Operation, the Dedicated Short Range Communication Working Group of the ITS Committee, December, 2008.

[10] Charles E. Perkins, Pravin Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, ACM SIGCOMM Computer Communication Review 24 (4) (1994) 234–244.

[11] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, July 2003.

[12] S.Y. Wang et al., The design and implementation of the NCTUns 1.0 network simulator, Computer Networks 42 (2) (2003) 175–197.

[13] S.Y. Wang et al., A vehicle collision warning system employing vehicle-to-infrastructure communication, in: IEEE Wireless Communications and Networking Conference 2008 (WCNC 2008), March 31–April 3, Las Vegas, USA, 2008.

[14] N. Ferreira, J.A. Fonseca, J.S. Gomes, On the adequacy of 802.11p MAC protocols to support safety services in ITS, in: Proceedings of the IEEE Emerging Technologies and Factory Automation, 2008 (ETFA 2008), Sept. 15–18, Hamburg, Germany, 2008, pp. 1189–1192.

[15] K. Bilstrup et al., Evaluation of the IEEE 802.11p MAC method for vehicle-to-vehicle communication, in: Proceedings of the IEEE 68th Vehicular Technology Conference, 2008 (VTC 2008-Fall), Sept. 21–24, Calgary, Alberta, Canada, 2008, pp. 1–5.

[16] Yunpeng Zang et al., Towards broadband vehicular ad-hoc networks – the vehicular mesh network (VMESH) MAC protocol, in: Proceedings of the IEEE Wireless Communications and Networking Conference, 2007 (WCNC 2007), 11–15 March 2007, Hong Kong, China, pp. 417–422.

[17] N. Choi et al., A solicitation-based IEEE 802.11p MAC protocol for roadside to vehicular networks, in: 2007 Mobile Networking for Vehicular Environments Workshop (MOVE 2007), May 11, Anchorage, Alaska, USA, 2007, pp. 91–96.

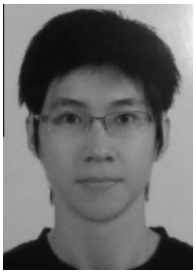
[18] C. Campolo, A. Cortese, A. Molinaro, CRaSCH: a cooperative scheme for service channel reservation in 802.11p/WAVE vehicular ad hoc networks, in: International Conference on Ultra Modern Telecommunications & Workshops, (ICUMT '09), Oct. 12–14, St. Petersburg, Russia, 2009, pp. 1–8.

[19] J.A. Ferreiro-Lage, C.P. Gestoso, O. Rubinos, F.A. Agelet, Analysis of unicast routing protocols for VANETs, in: Fifth International Conference on Networking and Services, 2009 (ICNS '09), April 20–25, Valencia, Spain, 2009, pp. 518–521.

Shie-Yuan Wang is a Professor of the Department of Computer Science at National Chiao Tung University, Taiwan. He received his Master and Ph.D. degree in Computer Science from Harvard University in 1997 and 1999, respectively. Before that, he received his Master degree in computer science from National Taiwan University in 1992 and his bachelor degree in computer science from National Taiwan Normal University in 1990. His research interests include wireless networks, Internet technologies, network simulations, and operating systems. He authors a network simulator, called NCTUns, which is now is a famous tool widely used by people all over the world.

Chih-Che Lin currently works for the Industrial Technology Research Institute of Taiwan. He received his BS degree and Ph.D. degree in Computer Science from National Chiao Tung University, Taiwan, in 2002 and 2010, respectively. He was a core team member of the NCTUns network simulator project during 2002 and 2010. His current research interests include wireless networking, wireless mesh networks, vehicular networks, intelligent transportation systems, and network simulation.

Wei-Jyun Hong currently works for the ASUStek Computer Incorporated. He received his BS degree and MS degree in Computer Science from National Chiao Tung University, Taiwan, in 2009. He was a core team member of the NCTUns network simulator project during 2008 and 2009. His major research interests include wireless vehicular networks.



Kuang-Che Liu currently works for the Wistron Corporation. He received his BS degree and MS degree in Computer Science from National Chiao Tung University, Taiwan, in 2008. He was a core team member of the NCTUns network simulator project during 2007 and 2008. His major research interests include wireless vehicular networks.