Evaluating and Improving the TCP/UDP Performances of IEEE 802.11(p)/1609 Networks

Shie-Yuan Wang, Hsi-Lu Chao, Kuang-Che Liu, Ting-Wei He, Chih-Che Lin and Chih-Liang Chou
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
Email: {shieyuan, hlchao, gjliou, anton, linjc, jlchou}@cs.nctu.edu.tw

***Abstract*—The IEEE 802.11(p)/1609 standard is an emerging technology for vehicular communication networks. It amends the IEEE 802.11-2007 standard and defines a new WAVE operational mode for vehicular environments. The WAVE mode utilizes a combined FDMA/TDMA scheme to manage network bandwidth. In this mode, a node must periodically switch its frequency channel between two different channels over time. Such a design may result in bandwidth wastage due to the residual time caused by channel switching. In this paper, we propose two easy-to-implement schemes to mitigate this problem and compare their TCP/UDP performances with those of the original scheme. Our simulation results provide many insights into the operations of these schemes on IEEE 802.11(p)/1609 networks and show that the proposed schemes always outperform the original scheme.**

I. Introduction

The IEEE 802.11(p) standard [1] is an emerging technology for vehicular communication networks. This standard aims to provide a solution for deploying vehicular communication networks based on IEEE 802.11-based radios. It amends the IEEE 802.11-2007 standard [2], defines a new operational mode for wireless accesses in vehicular environments (referred to as the WAVE mode), and cooperates with the IEEE 1609 standard family [3]. As shown in Fig. 1, 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 accommodates existing IP-based network applications and the latter is used to disseminate small packets that carry emergent safety, road, or traffic information.

The physical-layer specification of the IEEE 802.11(p)/1609 network is an amended version of the IEEE 802.11(a) physical-layer specification, which is an OFDM-based radio technology operating at 5.9-GHz frequency band. The IEEE 802.11(p)/1609 MAC layer manages link bandwidth in a combined FDMA/TDMA manner. Fig. 2 illustrates how an IEEE 802.11(p)/1609 network utilizes its bandwidth resource. As Fig. 2 shows, the used frequency spectrum is divided into one control channel (CCH) and several service channels (SCH). The CCH is dedicated for nodes to exchange network control messages while SCHs are used by nodes to exchange their data packets and WAVE-mode short messages.

The link bandwidth of these channels are further divided into transmission cycles on the time axis, each comprising a control frame and a service frame. These frames are represented by the black blocks and gray blocks, respectively in Fig. 2. In the draft standard defined in [3], it is suggested that the duration of a frame (either a control or a service frame) is set to 50 milliseconds. A footnote in the draft standard states that this value may be adjusted in the future standard, showing that different values may be used for different applications. In a transmission cycle, the control frame must be on CCH whereas the service frame can be on a specific SCH. The operation of the WAVE mode is briefly explained below.

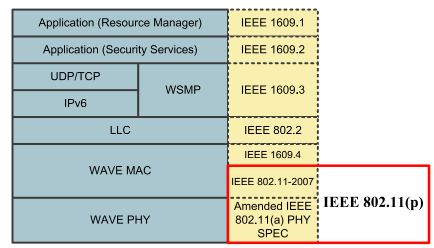


Fig. 1: The protocol stack of an IEEE 802.11(p)/1609 network

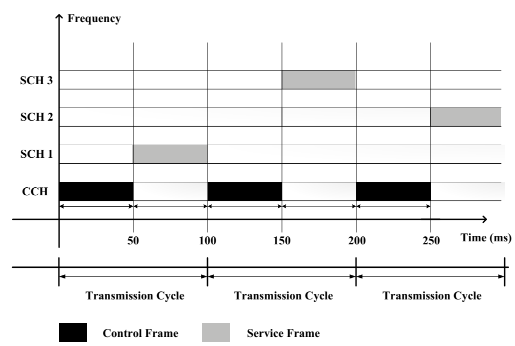


Fig. 2: The operation of an IEEE 802.11(p) network

After attaching to an IEEE 802.11(p)/1609 network, a node should first operate on CCH to gather necessary network information. In the WAVE mode, data packet transmissions are only allowed to occur within a Wave-mode Basic Service Set (WBSS). A node that initiates a WBSS is called a WBSS provider and nodes that join a WBSS are called WBSS users. To establish a WBSS, a WBSS provider has to periodically broadcast a WBSS announcement message (WAM) for this

978-1-4244-2703-1/08/$25.00 ©2008 IEEE

WBSS on CCH. A WAM includes the operational information of a WBSS (e.g., the ID of this WBSS and the SCH that this WBSS uses). A node should monitor all WAMs on CCH to know the existence and the operational information of available WBSSs. After knowing the SCH of the WBSS that it wants to join, a node may join this WBSS by periodically switching its channel to the SCH used by this WBSS on its service frames.

In the WAVE mode, WBSS users need not perform the authentication and association procedures to join a WBSS. (Note: these two procedures are necessary for a node to join an infrastructure BSS in the infrastructure mode of IEEE 802.11(a/b/g) networks.) The reason is that in a high-mobility environment such as a vehicular communication network, wireless link connectivity among vehicles is very fragile. In such a condition, the chance for a high-speed vehicle to join a WBSS is much smaller than a fixed/nomadic computer to join an infrastructure BSS. With this design, a vehicle can quickly utilize the bandwidth of a WBSS after detecting its existence. Because during the lifetime of a WBSS a WBSS provider may change the operational parameters of the WBSS, a WBSS user should switch back to CCH in every transmission cycle to learn the latest information about its WBSS.

The above CCH/SCH channel switching scheme results in a "bandwidth wastage" problem. As shown in Fig. 3, suppose that a packet transmission is going to take place (after the DIFS and Back-off time mandated by IEEE 802.11(a)) near the end of a service frame. The estimated transmission time (ETT) for this packet, however, exceeds the residual time of the current service frame. Because all nodes must switch back to CCH at the beginning of the following control frame, the receiving node cannot completely receive this packet with success. As such, the IEEE 802.11(p)/1609 draft standards recommend that in such a condition the transmitting node should prevent sending out this packet but instead should send it in the next service frame. Although this design avoids bandwidth wastage caused by incomplete packet reception, it results in bandwidth wastage at the end of each service frame due to not using the residual time. More details are explained below.

Assume that in an IEEE 802.11(p)/1609 network a node is serving a flow that continuously generates packets of 1400 bytes in length, and the data rate of a channel is fixed to 3 Mbps, which is the mandatory data rate that an IEEE 802.11(p) device must support. As such, the ETT of a 1400-byte data packet is 3.646 milliseconds. Suppose that the duration of a service frame is set to 50 milliseconds. In such a condition, if a node refrains itself from transmitting a packet near the end of a service frame due to the reason stated above, it will waste the channel bandwidth by 7.3% (3.646/50) in the worst case. Such bandwidth wastage will increase as the duration of a service frame decreases. For example, it will increase up to 18.23% if the frame duration decreases to 20 milliseconds. As stated above, the draft standard states that different durations can be used for IEEE 802.11(p)/1609 networks. Sometimes, it may be advantageous to use a shorter frame duration to achieve less end-to-end packet delays for real-time and emergent safety applications. In such applications, this bandwidth wastage problem can be significant.

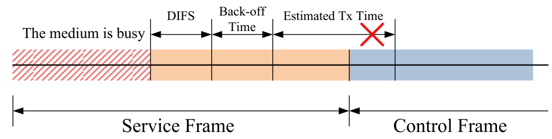


Fig. 3: The bandwidth wastage problem of an IEEE 802.11(p) network

To address this problem, in this paper we first evaluate the TCP/UDP performances of IEEE 802.11(p)/1609 networks under various network configurations. Then, we propose two easy-to-implement schemes to reduce bandwidth wastage caused by channel switching in an IEEE 802.11(p)/1609 network. To the best of the authors' knowledge, this is the first paper that studies the TCP/UDP performances of an IEEE 802.11(p)/1609 network and proposes schemes to address this bandwidth wastage problem. This paper provides many important insights into the operations of these schemes on IEEE 802.11(p)/1609 networks. The rest of this paper is organized as follows. In Section II, we discuss related work. We present our proposed schemes in detail in Section III and then evaluate the performances of these schemes in Section IV. Finally, in Section V we conclude the paper.

II. Related Work

In the literature, most of the papers (e.g., [4], [5]) focus on studying and solving issues of the IEEE 802.11(p) physical layer and thus are unrelated to our study. In [6], the authors evaluate the MAC-layer performances of an IEEE 802.11(p)/1609 network using an analytical and simulation approaches. The throughput, delay, and collision probability results under different node densities are presented in that paper.

In [7], the authors observe that the users of a WBSS are continuously changing over time due to the high-mobility property of vehicular networks. As such, 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 propose a new WBSS-user-oriented MAC-layer operational mode for the IEEE 802.11(p) standard. Using such an operational mode, a WBSS provider will transmit data packets to a WBSS user only when it receives a "transmission request" control message from the WBSS user.

Differing from the previous work, our paper focuses on studying the TCP/UDP performances of IEEE 802.11(p)/1609 networks and mitigating the bandwidth wastage problem caused by the channel-switching mechanism mandated by the IEEE 802.11(p)/1609 standards. In Section III, we propose two schemes to mitigate this problem.

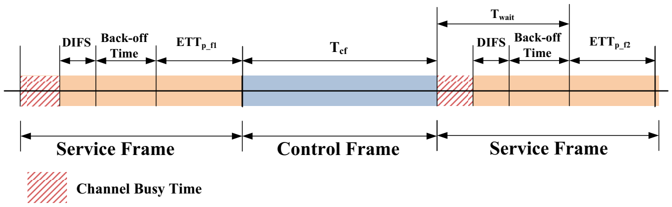


Fig. 4: The fragmentation scheme

III. The Proposed Schemes

In this section, we propose two easy-to-implement schemes to mitigate the bandwidth wastage problem presented in Section I: 1) the "fragmentation" scheme and 2) the "best-fit" scheme. The operations of these two schemes are presented in Section III-A and Section III-B, respectively.

A. The Fragmentation Scheme

As explained in Section I, for a transmitting node if the ETT of the first packet in its output queue exceeds the residual time of the current service frame, it should prevent transmitting this packet. To reduce the bandwidth wasted by this design, we propose a packet fragmentation scheme to utilize the residual time at the end of a service frame. Assume that a transmitting node is going to transmit a data packet p near the end of a service frame. As shown in Fig. 4, in the fragmentation scheme the transmitting node partitions p into two fragments p_f1 and p_f2. Based on the adopted data rate, the transmitting node determines the length of p_f1 so that the ETT of p_f1 is equal to the residual time of the current service frame. As such, there is no unused time (bandwidth) in a service frame, which mitigates the bandwidth wastage problem.

B. The Best-fit Scheme

The fragmentation scheme utilizes the residual bandwidth of a service frame at a small cost of one MAC-layer header and ACK packet for the second fragment (i.e., p_f2) of a fragmented packet. This scheme, however, does not reduce the end-to-end delays experienced by an application's packets. The reason is explained below. As Fig. 4 shows, at the end of a service frame the transmitting node partitions a large packet p into two packet fragments p_f1 and p_f2. Using the fragmentation scheme, the packet fragment p_f1 can be transmitted in the current service frame to the receiving node. However, the receiving node cannot deliver the received packet fragment p_f1 to the upper-layer application in the current service frame. Instead, it must keep p_f1 until it receives the second packet fragment p_f2 in the next service frame. Only at that time can it reassemble these two packet fragments into the original packet p and send p to the upper-layer application.

From this observation, one can formulate the delay experienced by a packet p using Equation 1.

$$Delay(p) = ETT_{p\_f1} + T\_cf + T_{wait} + ETT_{p\_f2} \quad (1)$$

where ETT_{p_f1} and ETT_{p_f2} denote the ETT values of p_f1 and p_f2, respectively, T_{cf} denotes the time duration of a control frame, and T_{wait} is a random variable denoting the time required by the transmitting node to access the wireless medium.

As explained previously, the standard in [3] currently suggests that T_{cf} is set to 50 milliseconds. Thus, the delay time of the packet p is at least 50 milliseconds. Such a large packet delay and variation may degrade the performances of delay-sensitive applications and protocols (e.g., VoIP and TCP), which require packets to be transmitted in a smooth and timely manner. One solution is to decrease the duration of a frame from 50 milliseconds to a smaller value. In Section IV, we conduct simulations with different frame durations to evaluate the impacts of this parameter on the performances of IEEE 802.11(p)/1609 networks.

On the other hand, in a multi-packet-size environment a transmitting node may have packets of different sizes in its output queue. Instead of fragmenting a large packet, a transmitting node can choose a smaller packet in the output queue and send it out, if the transmission time of this smaller packet is less than the residual time of the current service frame. This way, the transmitting node can reduce the delay experienced by the chosen packet (at least by T_{cf} milliseconds) while utilizing the residual bandwidth of the current service frame.

In this section, we propose the best-fit scheme to utilize the residual bandwidth of service frames and decrease the delays experienced by an application's packets. The operation of the best-fit scheme is described below. In the best-fit scheme, the transmitting node first checks whether the ETT of the first packet in the output queue exceeds the residual time of the current service frame. If not, it sends out the packet. Otherwise, it searches its output queue to examine whether there are packets with ETT values that are less than the residual time. If they exist, the transmitting node chooses the packet whose ETT value is closest to the residual time and sends it out. Otherwise, it does not send out any packet in the residual time. In the best-fit scheme, multiple smaller packets can be transmitted in the residual time to best utilize the residual bandwidth. This can be achieved by repeating the above processing until the latest residual time is so small that no more packet in the output queue can be further transmitted.

Fig. 5 illustrates the operation of the best-fit scheme on the time axis. As one sees, in a multi-packet-size environment, the best-fit scheme may be able to choose a suitable packet to transmit at the end of a service frame without fragmenting packets. As such, it improves the bandwidth utilization of service frames and decreases the delays experienced by an application's packets. One possible problem with this scheme is that packets of the same flow may become out of order when they arrive at the receiving node. However, TCP can reorder received packets to restore the original order. For applications that use UDP and are sensitive to packet reordering, they usually implement their own packet reordering functions inside themselves to handle possible packet re-orderings on IP networks (note: a network using the IP protocol does not guarantee in-order packet forwarding). For these reasons, these infrequent packet re-orderings (which may happen only at the

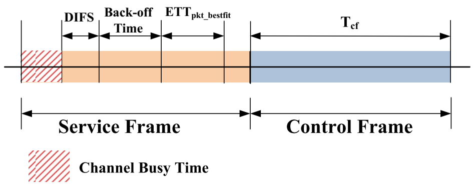


Fig. 5: The best-fit scheme

end of service frames) degrade performance insignificantly, as will be evidenced by our simulation results later.

IV. Performance Evaluation

In this section, we use the NCTUns network simulator [8] to evaluate the TCP/UDP performances of our proposed schemes and the original scheme defined in the IEEE 802.11(p)/1609 standards. The used TCP/UDP protocol stacks in NCTUns are the real-life TCP/UDP protocol stacks in the Linux kernel. Besides, the traffic generators used in NCTUns are real-life network applications using standard socket system calls to transmit/receive data over a simulated network. As such, the TCP/UDP throughput performances reported in this paper represent the performances of real-life TCP/UDP protocols over simulated IEEE 802.11(p)/1609 networks.

The important parameters used in our simulations are presented here. The simulated time of each case is set to 100 seconds. The data rate of each node's MAC layer is set to 3 Mbit/sec. The maximum transmission range of each node is set to 500 meters. The frame durations of the control and the service frame are varied from 20 to 50 ms.

We use two scenarios to study the performances of these schemes. One is the single source-destination-node-pair (SDNP) scenario (denoted as the SSDNP scenario) and the other is the multiple source-destination-node-pair scenario (denoted as the MSDNP scenario). In the SSDNP scenario, the simulated topology is a network composed of only two nodes, each of which is equipped with an IEEE 802.11(p) radio. The distance between them is set to 150 meters, which is less than the maximum transmission range. On the other hand, in the MSDNP scenario the simulated topology is a network having $2N$ IEEE 802.11(p)-capable nodes, where N is the number of different source-destination node pairs in the simulated network. In the MSDNP scenario, the distance between any pair of nodes is less than the maximum transmission range (i.e., 250 meters). As such, all nodes are within the same collision domain and may interfere with each other. Using such a configuration, we can evaluate whether the number of contending nodes in the network would affect the performances of the IEEE 802.11(p) protocol.

In our simulations, a flow on a SDNP may be unidirectional or bidirectional. A bidirectional UDP flow represents two unidirectional UDP flows operating in opposite directions on the same SDNP. Although a TCP connection is bidirectional (i.e., data can be transmitted in opposite directions on a TCP connection at the same time), in our simulations a TCP flow sends its data only in one direction on the TCP connection. As such, we call such a TCP flow a unidirectional TCP flow in this paper. Based on this definition, we use the term "a bidirectional TCP flow" on a SDNP to represent the two unidirectional TCP flows that operate in opposite directions on the same SDNP.

In a simulation case, the traffic type of all flows in the simulated network is the same – either greedy TCP or greedy UDP that continuously generates and transmits 1400-byte UDP packets. In the SSDNP scenario, only one flow is set up on the SDNP. In contrast, in the MSDNP scenario, when there exists multiple flows (either unidirectional or bidirectional) in the network, each flow is set up on a different SDNP. For example, suppose that a MSDNP scenario case has five bidirectional TCP flows. In this case, the network has five SDNPs and each of the five bidirectional TCP flows (i.e., two unidirectional TCP flows operating in opposite directions) is set up on a different SDNP.

For each simulation case, we conducted ten runs, each time using a different random number seed. The average of the results of these runs is then reported as the performance of the simulated case. In the following, we present the simulation results of the SSDNP and MSDNP scenarios.

A. The SSDNP Scenario

In the SSDNP scenario, the number of flows is fixed to one in all simulation cases. Fig. 6 shows the throughput of one unidirectional UDP flow over different frame durations. There are several findings about this figure. First, the fragmentation scheme outperforms the original and best-fit schemes over all frame durations and the maximum performance improvement is 16% (144/124) when the frame duration is 23 milliseconds. This result is expected, as the fragmentation scheme can always fragment a large packet to utilize the residual bandwidth of service frames. As such, it should perform the best. Second, the performances of the original and the best-fit schemes are exactly the same. This result is also expected. In this case, the size of all UDP packets are the same (i.e., 1400 bytes). In such a condition, the best-fit scheme degenerates to the original scheme because it cannot find any smaller packet to utilize the residual bandwidth of service frames. As such, the performances of the two schemes are exactly the same.

Third, the performance of the original (and the best-fit) scheme goes up and down as the frame duration decreases, and the difference between the "up" and "down" increases as the frame duration decreases. This result can be explained. The worst case for the original scheme is when the residual time is just slightly less than the transmission time of a 1400-byte UDP packet, because in such a condition the residual time cannot be utilized at all. The percentage of bandwidth wastage in this condition is about the transmission time of a 1400-byte UDP packet divided by the frame duration. According to this formula, when the frame duration decreases, the percentage of worst-case bandwidth wastage (i.e., the difference between the fragmentation scheme and the original scheme) will increase. As for the "up and down" behavior, the "down" points correspond to the worst cases described above while the "top" points correspond to the best cases to be described immediately. The best case occurs when the frame duration happens to be a multiple of the transmission time of a 1400-byte UDP packet, because in this condition no residual time is unused in a frame duration. Since in this case the traffic pattern is so regular and each packet has the same size, the effect of the "top" points is evident.

The aggregate throughput of one bidirectional UDP flow over different frame durations is almost the same as its unidirectional counterpart. As such, we do not present it here to save space. Recall that a bidirectional UDP flow represents two unidirectional UDP flows operating in opposite directions. The aggregate throughput of one bidirectional UDP flow is defined to be the sum of the throughputs of these two competing unidirectional UDP flows. In this bidirectional UDP case, although now there are two nodes with UDP packets to transmit (i.e., the two nodes of the SDNP now compete for the shared wireless bandwidth of service frames), the size of the UDP packets in their output queues are the same (i.e., 1400 bytes). As such, the best-fit scheme degenerates to the original scheme on each node, resulting in the same behavior observed in the unidirectional UDP case.

Fig. 7 shows the throughput of one unidirectional TCP flow over different frame durations. One sees that the relative performances of the three schemes are similar to those in the unidirectional UDP flow case. This phenomenon can be explained. In this case, one node (the source node of the TCP flow) continuously transmits large TCP data packets of the same size to the other node. The size of these large data packets is the "Maximum Transmission Unit" of the network interface and is normally 1500 bytes. On the other hand, under the control of the TCP congestion and error control protocols, the other node continuously transmits small (40 bytes) TCP ACK packets back to the source node when TCP data packets are received. Because on each node the size of the packets in its output queue is the same (either 1500 bytes or 40 bytes), the best-fit scheme degenerates to the original scheme on each node, which explains why the original and the best-fit schemes have similar performances. From this figure, one sees that the "up and down" behavior still exists but the difference between the "up" and "down" points is smaller than that in the unidirectional UDP flow case. This phenomenon is explained below.

In this case, the two nodes compete for the shared wireless bandwidth and the sizes of their packets are different. As such, the packets transmitted on the service frames are a mix of large TCP data packets and small TCP ACK packets, affected by TCP congestion and error control protocols. Due to this randomness, the transmission timing of these packets on service frames is not as regular as that in the unidirectional UDP flow case. This means that the residual time in each service frame is not always the maximum time in the worst case but sometimes is smaller. As such, the difference between the "up" and "down" points is smaller than that in the unidirectional UDP flow case.

Fig. 8 shows the aggregate throughput of one bidirectional TCP flow over different frame durations. Recall that one

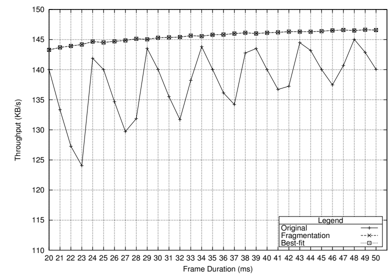


Fig. 6: The throughput of one unidirectional (or one bidirectional) UDP flow over different frame durations

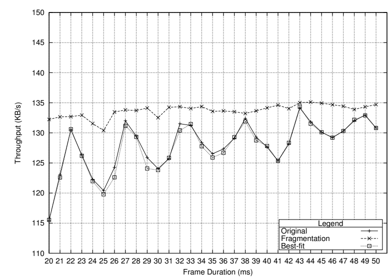


Fig. 7: The throughput of one unidirectional TCP flow over different frame durations

bidirectional TCP flow is composed of two unidirectional TCP flows and the aggregate throughput of a bidirectional TCP flow is defined as the sum of the throughputs of its two unidirectional TCP flows. Compared with Fig. 7, there are two new findings.

First, the best-fit scheme now outperforms the original scheme and performs almost as well as the fragmentation scheme. This change can be explained. In this case, each of the two nodes has large TCP data packets and small TCP ACK packets mixed in its output queue to send. Due to this packet size diversity, the best-fit scheme can find small TCP ACK packets in the output queue to utilize the residual time in the service frames. Because the size of a TCP ACK packet is small (only 40 bytes), the residual time that still cannot be used in the best-fit scheme is reduced to a value less than the transmission time of a small TCP ACK packet. This explains why the best-fit scheme now performs almost as well as the fragmentation scheme.

Second, the "up and down" behavior of the original scheme in the unidirectional TCP (and unidirectional UDP) flow case is no longer evident in this bidirectional TCP flow case. The reason for this change is presented below. Since now the output queue of each node includes large TCP data packets and small TCP ACK packets mixed together, even without the help of the best-fit scheme, the transmission timing of packets is no longer as regular as that in the unidirectional TCP (and unidirectional UDP) flow case. As such, the chance that the residual time in every service frame happens to be the maximum in the worst case is very tiny. In addition, even if on one node some large residual time is left, this residual time may be utilized by the other node to transmit some small TCP ACK packets

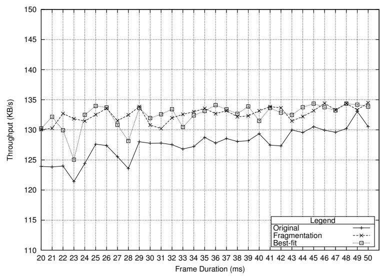


Fig. 8: The aggregate throughput of one bidirectional TCP flow over different frame durations

that happen to be at the head of its output queue. For these reasons, the original scheme does not exhibit the "up and down" behavior in the bidirectional TCP flow case.

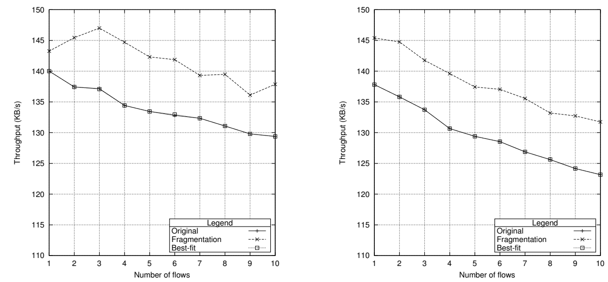
B. The MSDNP Scenario

Here we study the performances of the three schemes under the MSDNP scenario. In this scenario each flow is set up on a different SDNP and the reported throughput is the total throughput of these competing flows. Fig. 9a and Fig. 9b show the total throughput of multiple unidirectional and bidirectional UDP flows, respectively. The findings that we have obtained from the SSDNP scenario can explain these figures as follows. In this case, all UDP packets are of the same size and as such, the best-fit scheme degenerates to the basic scheme on each of these nodes competing for the shared wireless bandwidth. This is why (1) the fragmentation scheme outperforms the original and the best-fit schemes and (2) the original and best-fit schemes have similar performances. One new finding is that as the number of competing flows (nodes) increases, the total throughput decreases. This phenomenon is caused by the property of the CSMA/CA protocol, which IEEE 802.11(p) uses.

Fig. 10a and Fig. 10b show the total throughput of multiple unidirectional and bidirectional TCP flows, respectively. They can be explained using the findings obtained before. For example, one sees that the three schemes perform almost the same when the number of flows is large. This is due to the packet size diversity and the fact that there are multiple nodes competing for the shared wireless bandwidth. With multiple competing nodes, if there is residual time on one node, it may be utilized by another node to transmit a small TCP ACK packet. When the number of competing flows (nodes) increases, the probability that there is a node that has a small TCP ACK packet to utilize the residual time increases as well. This explains why the performances of the three schemes are about the same in this case.

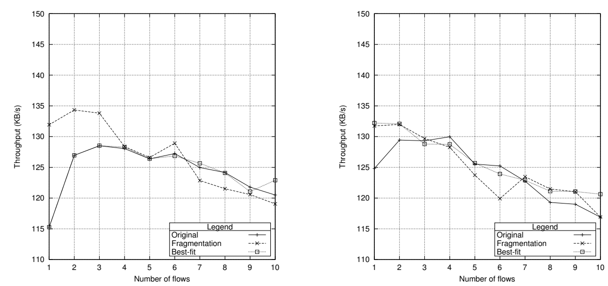
V. Conclusion

In this paper, we evaluate the TCP/UDP performances of IEEE 802.11(p)/1609 networks under various network configurations. We first point out the bandwidth wastage problem caused by channel switching mandated in the IEEE 802.11(p)/1609 standards. To mitigate this problem, we propose two easy-to-implement schemes called the fragmentation and the best-fit schemes and then compare their TCP/UDP performances with those of the original scheme under various network configurations. Our simulation results provide several important insights into the operations of these schemes on IEEE 802.11(p)/1609 networks.



(a) Unidirectional UDP flows (b) Bidirectional UDP flows

Fig. 9: The total throughput of multiple UDP flows with the 20-ms frame duration



(a) Unidirectional TCP flows (b) Bidirectional TCP flows

Fig. 10: The total throughput of multiple TCP flows with the 20-ms frame duration

References

[1] "IEEE 802.11p/D3.0," IEEE Standards Activities Department, July 2007.

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

[3] "IEEE 1609.4 Trial-Use Standard for Wireless Accesses in Vehicular Environments (WAVE) - Multi-channel Operation" IEEE Vehicular Technology Society, October 2006.

[4] Weidong Xiang et al., *"Introduction and Preliminary Experimental Results of Wireless Access for Vehicular Environments (WAVE) Systems,"* 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops 2006, p.p. 1-8, July 17-21, 2006, San Jose, CA, USA.

[5] L. Stibor et al., *"Evaluation of Communication Distance of Broadcast Messages in a Vehicular Ad-Hoc Network Using IEEE 802.11p,"* IEEE Wireless Communications and Networking Conference 2007, (WCNC 2007), p.p. 254-257, March 11-15, 2007, HongKong, China.

[6] Stephan Eichler, *"Performance Evaluation of the IEEE 802.11p WAVE Communication Standard,"* Proceedings of 2007 IEEE 66th Vehicular Technology Conference (VTC-2007 Fall), p.p. 2199-2203, Sept. 30 2007-Oct. 3 2007, Baltimore, MD, USA.

[7] Choi, N. et al., *"A Solicitation-based IEEE 802.11p MAC Protocol for Roadside to Vehicular Networks,"* 2007 Mobile Networking for Vehicular Environments Workshop (MOVE 2007), p.p. 91-96, 11 May, 2007, Anchorage, Alaska, USA.

[8] S.Y. Wang et al., *"NCTUns 4.0: An integrated simulation platform for vehicular traffic, communication and network researches,"* 1st IEEE WiVec 2007 (International Symposium on Wireless Vehicular Communications), September 30-October 1, 2007, Baltimore, MD, USA. (available for download at http://NSL.csie.nctu.edu.tw/nctuns.html.)