A MAC Protocol for Disseminating Loss and Delay Sensitive Messages in Wireless Packet Networks *

Jun Peng and Liang Cheng
Laboratory Of Networking Group (LONGLAB, http://long.cse.lehigh.edu)
Department of Computer Science and Engineering
Lehigh University
19 Memorial Drive West
Bethlehem, PA 18015, USA
{jpeng,cheng}@cse.lehigh.edu

Abstract

Existing medium access control (MAC) protocols for wireless packet networks are optimized for unicast flows in general and do not address the reliability and latency issues facing broadcast packets in the MAC sub-layer. Applications that rely on disseminating loss and delay sensitive (LDS) messages in wireless networks are therefore hindered by such protocols. This paper proposes a new MAC protocol for LDS message dissemination in wireless networks. The proposed protocol solves the hidden terminal problem for LDS messages and also provides multiple levels of strict priority for them in medium access. Extensive simulations showed the effectiveness of the proposed MAC protocol in supporting LDS message dissemination in ad hoc networks.

Keywords: ad hoc networks, medium access control, hidden terminals, and priority scheduling

1 Introduction

Existing MAC protocols for wireless packet networks, including the popular IEEE 802.11 MAC protocol [2], are designed for unicast flows in general and do not address the difficulties facing broadcast packets for medium access. On the other hand, some applications in wireless packet networks, such as safety/emergency message dissemination in vehicular ad hoc networks [3], require lossless and timely medium access for their broadcast packets. To design a new MAC protocol for such applications is therefore necessary.

Medium access control in wireless networks faces multiple difficulties. One of them is the well-known hidden terminal problem [4]. Basically, in a wireless network, two nodes that cannot hear each other may still interfere with each other at their receivers and cause losses. In addition, wireless links usually have higher loss rates than wired links. Another difficulty for a wireless MAC protocol is to support quality of service (QoS) in the MAC sub-layer because it is usually necessary for a MAC protocol for wireless packet networks to be fully distributed.

The IEEE 802.11 MAC protocol [2] and other similar protocols such as [5, 6, 7] use a technique dubbed "virtual carrier sense" to deal with the hidden terminal problem for unicast packets. This technique is, in general, not suitable for broadcast packets because of the difficulties and overhead in coordinating multiple receivers. The automatic repeat request (ARQ) technique used by IEEE 802.11 and other protocols to ensure reliability for unicast packets faces similar difficulties in the broadcast case. In addition, the service differentiation provided by IEEE 802.11e and other similar protocols such as [8, 9, 10, 11] is for statistic priority only and not suitable for applications that require strict priority in medium access, such as safety/emergency message dissemination in vehicular ad hoc networks.

Instead of relying on in-band control messages, some other MAC protocols such as [12, 13] use an out-of-band control channel to support priority scheduling. These protocols, however, focus on improving the performance of some *unicast* flows in congested situations, while broadcast packets receive not much attention. Additionally, these protocols do not address the hidden terminal problem, particularly for broadcast packets. An out-of-band control channel has also been used to suppress hidden terminals by some other existing protocols such as [4, 14, 15]. However, these protocols do not address the problem of priority scheduling in the MAC sub-layer. Therefore, no existing MAC protocol has the full capa-

---

* The basic idea of the work was published in [1] and this research has been financed by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA). Please direct all correspondence to Prof. Liang Cheng. Tel. (610) 758-5941, Fax (610) 758-4096, Email: cheng@cse.lehigh.edu.

bilities to support the type of application that requires the dissemination of loss and delay sensitive (LDS) messages in wireless packet networks.

This paper presents a new MAC protocol for wireless packet networks to support applications that require the dissemination of LDS messages, such as the safety/emergency messages in vehicular ad hoc networks. Using a novel pulse-based mechanism, the proposed MAC protocol realizes both hidden-terminal suppression and priority scheduling in the MAC sub-layer with a single narrow-bandwidth control channel. In a fully distributed way, the proposed protocol introduces preemptive priority scheduling into the MAC sub-layer and supports multiple levels of strict priority for LDS messages. The detailed simulation results in the paper show that the proposed protocol is free of the hidden terminal problem and at the same time ensures multiple levels of strict priority for LDS messages in ad hoc networks.

The rest of the paper is organized as follows. Section II presents in detail the proposed MAC protocol for disseminating LDS messages in wireless packet networks. Section III evaluates the proposed protocol with extensive simulations on NS-2. Finally, Section IV summarizes the paper.

2 The Proposed MAC Protocol

The basic approach of the proposed MAC protocol in this paper is to use "pulses" in an out-of-band control channel to suppress hidden terminals and do priority scheduling for LDS messages at the same time. "Pulses" in the proposed protocol are basically single-tone waves with pauses of *random* lengths. In the proposed protocol, the control channel carries *only* pulses and pulses appear *only* in the control channel. The control channel is monitored by all nodes all the time except when they are transmitting in the channel (an antenna usually cannot transmit and listen at the same time). A node that is generating pulses in the control channel still monitors the channel when its pulses pause.

With the proposed protocol, the application layer determines the priority level of a LDS message and put this information in the packet header [1]. The application layer also determines the number of duplicate copies to send for each LDS message. To send duplicate copies of a LDS message is for dealing with the problem that ARQ cannot be used to ensure reliability for LDS packets in the MAC sub-layer. On one hand, coordinating and sequencing multiple receivers to send back acknowledgments are difficult and also introduce extra delays. On the other hand, precise neighbor information may not be available when nodes are mobile in a wireless packet network. Another assumption of the proposed protocol is that there is a co-existing MAC protocol for normal packets, such as IEEE 802.11. LDS messages access the medium through the proposed MAC protocol, while normal packets do so through the co-existing protocol.

The rest of this section presents the proposed MAC protocol in detail. Subsection A introduces the pulses and priority scheduling in the proposed protocol. Subsection B describes the proposed protocol with pseudocode. Finally, Subsection C clarifies some important issues related to the design of the proposed protocol.

2.1 Pulses and Priority Scheduling

When a node is using the data channel for disseminating LDS packets, it continuously transmits pulses in the control channel, as shown in Fig. 1. The node stops its pulses only after its transmission of LDS packets is finished or other LDS message sources interrupt it.

Each pulse consists of an active part of a fixed length and a pause part of a *random* length, as shown in Fig. 2. In the active part, single-tone waves are transmitted in the control channel, while no wave is transmitted in the pause part. The random pause part is further divided into two sub-parts, a contention window of a fixed size and a residual pause of a random length. Moreover, the contention window is cut into equal-size sub-windows.

The active part of a pulse plays two roles. One is to suppress hidden terminals; any node hearing a pulse aborts its transmission. The other role is to deliver the priority level information for the LDS message in transmission. In particular, the length P_a of the active part of a pulse indicates the priority level L_r of the LDS message in transmission; a longer active part indicates a higher level of priority for the message in transmission.

$$L_{r1} > L_{r2} \quad if \ P_{a1} > P_{a2}$$

The priority level information L_r expires at a node if there has been no pulse in the control channel for a specified amount of time.

The random pause part of a pulse is to support multiple levels of priority for LDS messages. When LDS message sources S_1 to S_n detect a busy control channel but find that the priority levels L_1 to L_n of their messages are higher than the priority level L_r of the message in transmission, each of these sources starts a random backoff timer as soon as the pulse in the control channel pauses. The random backoff delay d_i of the source i is drawn in a contention sub-window that is determined by the priority level L_i of the source's LDS message. Basically, a higher level of priority acquires a lower sub-window and thus a shorter random delay.

$$d_j < d_k \quad if \ L_j > L_k$$

---

[1] "Message" and "packet" are used exchangeably in this paper because a LDS message is usually short and can be included in one packet.

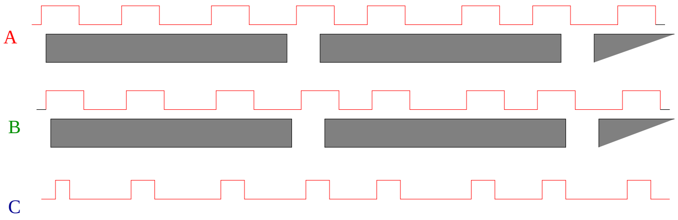

Figure 1: LDS Packets and Their Accompanying Pulses. Node A Is the LDS Message Source; Node B Is a Neighbor of Node A; Node C Is a Hidden Terminal of Node A.

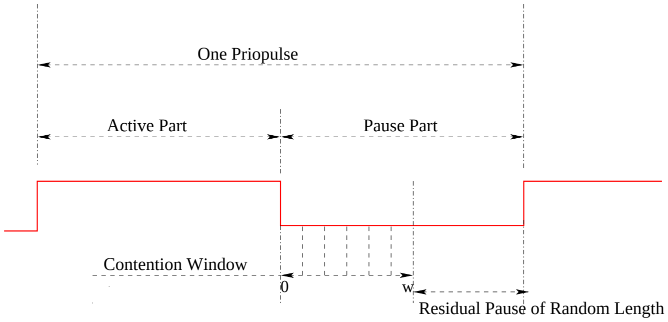

Figure 2: A Pulse Consists of an Active Part of a Fixed Length and a Pause Part of a *Random* Length. The Random Pause Part Is Composed of a Contention Window and a Residual Random Pause. Furthermore, The Fixed-Size Contention Window Is Cut Into Sub-Windows.

The source with the shortest backoff delay (i.e., of the highest level of priority) usually acquires the medium before other sources do, and this source becomes the winner source S_v in this round of contention.

As soon as the backoff timer of the winner source S_v expires, source S_v starts to transmit pulses in the control channel. On the other hand, the LDS message source S_o then owning the channels is still in its pause in the control channel because random backoff delays d_1 to d_n are drawn in the contention window and there is still a residual random pause behind it in the pulse of S_o. Source S_o therefore can detect the pulse of the winner source S_v and releases both channels.

However, two sources of the same level of priority, such as sources i and j, may draw similar delays in the same contention sub-window, which may result in a collision. The random length of the residual pause of a pulse is designed to deal with this problem. If sources i and j draw similar backoff delays in contending for the medium, the active parts of their first pulses are synchronized with each other and neither node will be aware of the other. However, the random-length pauses in their pulses will desynchronize the active parts in their pulses. After the desynchronization, one source will detect a pulse of the other and backs out to wait for the next chance to contend for the medium. After one source backs out, the other source will be able to successfully deliver its remaining message or messages.

2.2 The Scheme and Pseudo Code

This subsection presents the proposed protocol in a complete picture with pseudo code. Algorithm 1 presents how a LDS message source proceeds to access the medium with pulses, while Algorithm 2 shows how a node behaves with the proposed protocol. Most parts of the two algorithms are self-explanatory. The rest of this subsection explains how pulses are "relayed" by message receivers to suppress hidden terminals.

Pulses are "relayed" by nodes to suppress the hidden terminals of a LDS message source. Ideally, only the neighbors of a LDS message source should "relay" pulses. However, pulses, unlike packets, cannot contain address and hop-count information. Before introducing the mechanism in the proposed protocol to approach the ideal situation, we first introduce how a node "relays" a pulse.

Basically, a node "relays" a pulse to be a slightly shorter but "synchronized" pulse. When a node relays a pulse, it starts its relayed pulse as soon as the original one is detected (i.e., the relaying node does not wait for the reception of the whole original pulse). The active length p_a of the relayed pulse is, however, slightly shorter than

P_a, which is the active length of the original pulses in the control channel. The shorter active length of a relayed pulse is to prevent the original pulse source from hearing the relayed pulse (please notice that when a node starts to transmit in the control channel, it cannot hear any other transmissions in the channel until its own transmission finishes). If the active length P_a of an original pulse is unknown, then the relayed pulse only takes a short active length so that it will not be heard the original pulse source. This short-active-length pulse is generated to suppress hidden terminals and does not encode any emergency level information (i.e., its active length is not in the coding book). An example of LDS packets accompanied by pulses is shown in Fig. 1.

To approach the ideal situation of pulse relay, a node in the proposed protocol relays a pulse only in the following two cases. The first is that the node is busy receiving in the data channel *and* the pulse is the first detected one after the control channel became idle last time. The second case is that the node is receiving a LDS packet in the data channel.

The pulse relay in the first case is to clear the data channel for the incoming LDS packet. When the first pulse of a LDS message source arrives at its neighbors, they can not determine if they are the neighbors of an LDS message source, since pulses may be relayed but carry no address or hop count information. However, if some of these neighbors are receiving any packets, they need to stop their senders so that the LDS packet will arrive at them with a clear channel shortly after. Therefore, a node with the proposed protocol always relays a pulse if the pulse is the first one that the node detects in an idle control channel *and* meanwhile the node is busy receiving in the data channel. If such a node receives an LDS packet shortly after, it continues to relay the following pulses. Otherwise, the node will stop relaying the pulses.

Algorithm 1: A LDS Message Source Accessing the Medium with pulses

```
Start random backoff timer;
if Detect a pulse then
    if Backoff timer is busy then
        Cancel backoff timer;
    end
    Abort any transmission;
end
if Backoff timer expires then
    Start to transmit pulses;
    Take a short delay;
    Start to transmit LDS packets;
end
if No more LDS packets to transmit then
    Stop transmitting pulses;
end
```

Algorithm 2: A Node With the Proposed Protocol

```
repeat
    if Detect a pulse then
        switch State in data channel do
            case Busy Transmitting
                Abort transmission;
            case Busy Receiving
                if Receiving a LDS packet then
                    if Not in NO-RELAY state then
                        Relay the pulse;
                    end
                else
                    if First pulse in idle control channel then
                        Generate a short pulse;
                    end
                end
            case Idle
                Do nothing;
        end
        Measure the active length of the pulse $P_a$;
        Decode the priority level information $L_r$;
    end
    if A LDS packet comes from upper layer then
        if Busy control channel then
            if Legitimate $L_r$ && $L_x > L_r$ then
                Wait for a pause in the control channel;
                Access the medium with pulses;
            else
                defer;
            end
        else
            Access the medium with pulses;
        end
    end
    if A normal packet comes from upper layer then
        if Busy control channel then
            defer ;
        else
            Access the medium by IEEE 802.11;
        end
    end
    if Control channel becomes idle then
        if Having a deferred LDS packet then
            Access the medium with pulses;
        end
        if Having a deferred normal packet then
            Access the medium by IEEE 802.11;
        end
    end
    if Finish receiving a LDS packet then
        Send the packet to upper layer;
        if Receiving no more LDS packet then
            Stop relaying pulses;
        end
    end
    if Finish receiving a normal packet then
        Send the packet to upper layer;
    end
until Forever;
```

Figure 3: Illustration for NO-RELAY State.

The final restriction on pulse relay is that when a node has already been receiving pulses for a specified amount of time (50 μs in our implementation) but receives no LDS packet, the node enters the "NO-RELAY" state, in which it will not relay any pulses even if it starts to receive a LDS packet. The "NO-RELAY" state expires only after the control channel becomes idle again. This "NO-RELAY" design is to avoid the possible interference between two LDS message sources that cannot hear each other's pulses.

A simple example is shown in Fig. 3. If node A in this example is disseminating LDS packets, its neighbor node B will be relaying its pulses. Node C will thus sense pulses in its control channel but not relay them because it is not receiving an LDS packet. Therefore, Node D will not sense pulses. In such a case, node D may start to disseminate its own LDS packets. After node C starts to receive LDS packets from node D, it will start to relay pulses in the control channel. Consequently, node B and node C will relay pulses from each other. Since the pulses of node A and node D are not synchronized, either node A or node B, or both in the worst case, will be stopped by the pulses of the other node. However, if "NO-RELAY" state is established at node C in this example, node C will not relay any pulses until the control channel becomes idle again. The interference between nodes A and D is therefore resolved.

2.3 Other Design Considerations

This subsection introduces other issues that are related to the design of the proposed MAC protocol for LDS message dissemination in wireless packet networks. These issues are mainly related to pulses and the impact of propagation on them.

One requirement on pulses is that a pulse should have a length that is comparable to the transmission time of a LDS packet. Pulses are mainly designed for sources of higher levels of priority to interrupt a transmitting source of a lower level of priority (i.e., for the support of multiple levels of strict priority for LDS messages). Pulses therefore should be repeated at a frequency that makes it feasible for a transmitting source to be stopped before it finishes its transmission.

As introduced earlier, the active length of a pulse usually encodes the priority level information of the LDS message in transmission. In the proposed protocol, the active lengths used for encoding different levels of priority are usually significantly different (the difference is 100 μs in our implementation). This significant difference is to ensure the right decoding of priority level information after pulses experience propagations and relays.

Another phenomenon that may impact on pulse spreading is multipath fading, which occurs when a signal reaches a receiver via multiple paths. Multipath is a common phenomenon in urban areas due to obstacles and reflectors. Multipath may cause fluctuating amplitude and phase in signals, which is harmful for signal decoding. Pulses, however, are not as sensitive to multipath fading as bit-based packets. First, a pulse has a much longer transmission duration than a bit in a packet. For example, if there are 5 pulses in the transmission duration of a packet of a length of 100 bytes, then each pulse has an equal length of 160 bits in transmission time. Second, only amplitude fluctuation has significant impact on pulse detection.

The last thing to mention in this subsection is that the proposed MAC protocol is fully distributed and does not require nodes to be synchronized. When a group of receivers of a LDS message relay pulses in the control channel, they start their relayed pulses as soon as they detect the appearance of a pulse in the control channel. The original pulses from the LDS message source thus play the role of synchronizing the LDS message receivers for them to relay pulses. Additionally, pulses usually only go two hops, since only nodes that are receiving LDS packets relay all pulses.

3 Scheme Evaluation

3.1 Simulation Configurations

This section evaluates the proposed MAC protocol with extensive simulations. The proposed protocol was named "PreempPrio-MAC" because of the preemptive priority service that it provides to LDS packets in the MAC sublayer. The simulations were conducted on NS-2 [16] for the scenarios of LDS message dissemination in mobile ad hoc networks.

The implemented PreempPrio-MAC protocol in our simulations supports three levels of priority for LDS messages. The active lengths of pulses are 100, 200, and 300 μs, respectively, for encoding the three levels of priority. The contention window in each pulse has a size of 150 μs and each sub-window is 50 μs long. The residual pause window in which each pulse draws its residual random pause is 100 μs.

Another important detail of our simulations is that we used "blank" broadcast packets of small intervals to simulate pulses in the control channel. "Blank" means that these packets carry no address or other information. When a node receives a blank packet at the right power level (i.e., above the carrier sense threshold) in the control channel, it detects a pulse signal, while the active length of the pulse is "measured" as the time segment during which blank packets are continuously flowing in.

To show the performance of the approach that uses in-band control messages, we also simulated with the IEEE 802.11 and 802.11e MAC protocols. In IEEE 802.11e, the minimum backoff window limit for LDS packets is half of that for normal packets. Meanwhile, the AIFS for LDS packets is one timeslot less than that for normal packets.

In our simulations, fifty nodes move in an area of 500 by 500 meters. The wireless links of these nodes have a basic rate of 1 Mb/s. The carrier sense distance of each node in the data channel is about 300 meters, while the transmission distance is about 150 meters. The control channel uses the same power as the data channel to ensure pulse coverage. Each simulation lasts for 100 seconds, in which nodes may disseminate bursts of LDS packets to their neighbors to assess medium access delay and packet losses. In each burst, there are an average of 5 LDS packets and each packet has a size of 50 bytes. In addition, twenty five CBR flows of normal packets are randomly initialized in the background in each simulation.

3.2 Competition with Normal Traffic

This subsection shows how LDS traffic competes with normal traffic with the proposed MAC protocol and IEEE 802.11 MAC protocols. We conducted a series of simulations in which the packet intervals of background traffic were 1.0, 0.2, 0.04, 0.008, and 0.0016 second, respectively. Node movement was random waypoint in our simulations and the average pause time was 0.5 second. In addition, node minimum and maximum speeds were 5.0 and 10.0 meters per second, respectively. In each simulation, a node disseminated bursts of LDS packets to assess medium access latency and packet losses.

For LDS message dissemination, it is important to know the worst performance case for a protocol because each delayed or lost LDS message may have a serious consequence such as a fatal collision on highway. The maximum medium access delay for the packets in each burst is therefore one of the criteria that we use to evaluate each protocol. The average of the maximum medium access delays in bursts is shown in Fig. 4 for each case of network load (For easy reading, packet intervals in our figures are converted to packet rates, which determine network load). As shown in the figure, PreempPrio-MAC has a almost constant medium access delay of 1 ms for LDS packets. IEEE 802.11e, however, shows long medium access delays for LDS packets when background traffic load is significant, although it obviously outperforms IEEE 802.11, which provides no priority scheduling for LDS packets. The average of the maximum delays in bursts with IEEE 802.11e reaches 20 ms in the case in which the packet interval of background traffic is 0.008 second. We show more details for this case in Fig. 5.

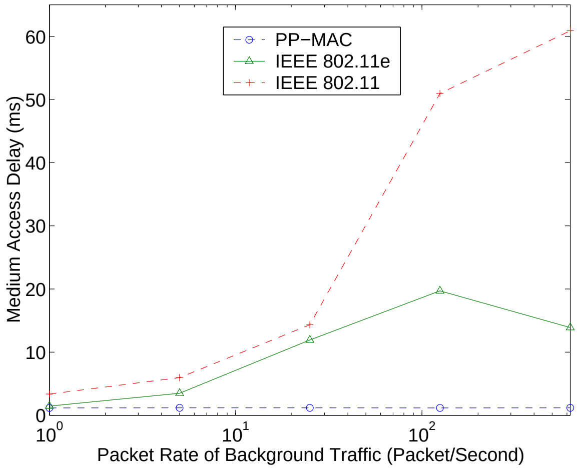

Figure 4: The Average of The Maximum Medium Access Delays in Bursts Against Background Traffic Load.

Fig. 5 shows the maximum medium access delay for the packets in each burst for the case in which the packet interval of background traffic is 0.008 second. As shown in the figure, PreempPrio-MAC has consistently low delays for LDS packets. IEEE 802.11e, however, shows long medium access delays for LDS packets, although its performance is much better than IEEE 802.11's. In the fifth burst, the maximum medium access delay for LDS packets reaches as high as 120 ms with IEEE 802.11e.

Fig. 6 shows the average of the minimum numbers of receivers for packets in bursts in each case of background traffic load. The results in the figure are averages and therefore not necessarily integers. As shown in the figure, PreempPrio-MAC does not have packet losses in any case of traffic load in the network, which means there are no hidden terminals with PreempPrio-MAC. IEEE 802.11 and 802.11e, however, show packet losses as network load increases to a specific level. When network load is light, hidden terminals may not cause problems to the network, since they have no packet to send most of the time. When network load becomes heavier, each hidden terminal has an increased chance to cause collisions in the network due to their increased demand for using the medium. The results in Fig. 6 demonstrate that IEEE 802.11 and 802.11e have the hidden terminal problem, which is intolerable for LDS message dissemination in wireless networks.

Interestingly, IEEE 802.11e has better performance than IEEE 802.11 on average, as shown in Fig. 6. For further details, we checked how LDS packets in bursts experienced losses in each case of network load. The results are shown in Fig. 7 for the case in which the packet in-

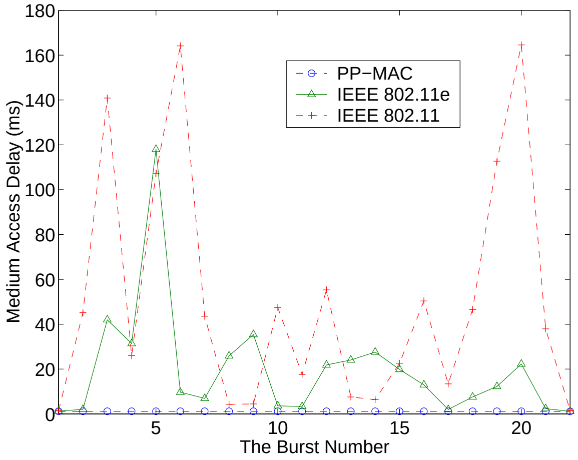

Figure 5: The Maximum Medium Access Delay for LDS Packets in Each Burst in A Single Case of Background Traffic Load.

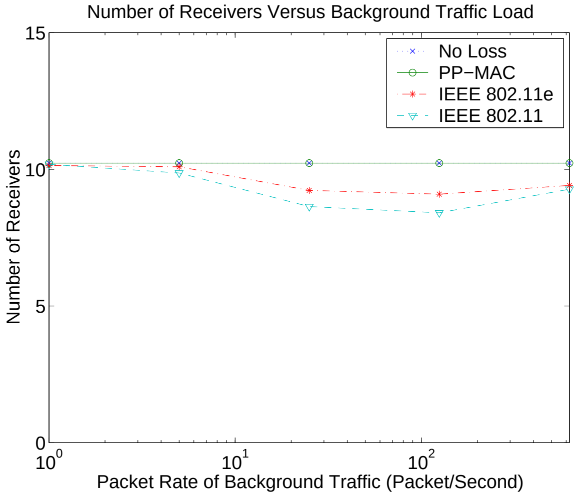

Figure 6: The Average of The Minimum Numbers of Receivers for Packets In Bursts In Each Case of Background Traffic Load.

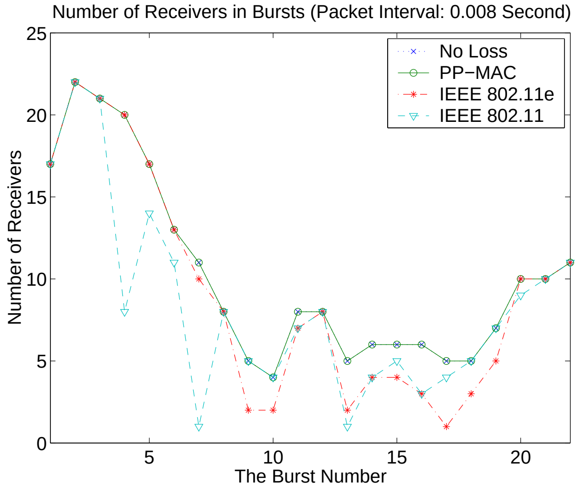

Figure 7: The Minimum Number of Receivers for Packets in Each Burst in A Single Case of Background Traffic Load.

terval of background traffic is 0.008 second. As shown in the figure, PreempPrio-MAC shows no loss. With both IEEE 802.11e and 802.11, there are about 11 bursts that experience losses, which is a loss rate of about 50 percent considering that the total number of bursts is 22. However, IEEE 802.11, on average, has severer losses in bursts than IEEE 802.11e, as shown in Fig. 7.

These results are seemingly surprising since IEEE 802.11e does not have specific improved mechanisms to deal with hidden terminals as compared to IEEE 802.11. The performance discrepancy between them, in fact, comes from the "capture" phenomenon in wireless networks. With a "capture", a packet can still be successfully decoded when it collides with another packet of lower power. In the code implemented in NS-2, a capture only happens to a packet that is the first one arriving at an idle node, which reflects the design of some hardware. With its lower backoff window limit and shorter AIFS in the IEEE 802.11e case, a LDS message source usually starts to transmit earlier than a normal packet source if they are hidden terminals to each other and start to contend for the idle medium simultaneously. Therefore, captures happen more often to LDS packets in the IEEE 802.11e case than in the IEEE 802.11 case, which explains the performance discrepancy between the two protocols shown in Fig. 6 and Fig. 7.

Another interesting thing shown in Figs. 4 and 6 is that the performance of IEEE 802.11 and 802.11e does not necessarily decrease for LDS packets as network load increases. Instead, their performance for LDS packets may even improve after network load reaches a specific level. With high network load, medium contentions

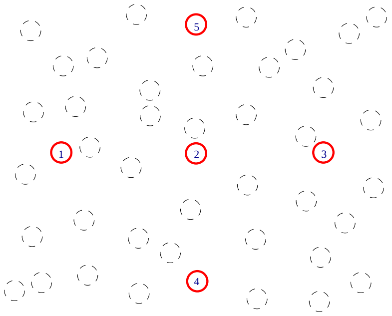

Figure 8: The Locations of the Five Competing LDS Message Sources.

among nodes become severe, which causes long backoffs in medium access. With longer backoffs, a hidden terminal has a lower chance to cause collisions, which explains why Fig. 6 shows improved performance for IEEE 802.11 and 802.11e after network load reaches a specific level. In addition, LDS packets may specifically benefit from high network load in the IEEE 802.11e case. With network load reaching a specific level, normal packets compete among themselves severely and cause long backoffs to them, which benefits LDS packets due to their lower backoff limits. Fig. 4 therefore shows a lowered delay for LDS packets in the IEEE 802.11e case when the network load goes beyond a specific level.

3.3 Competition Among LDS Message Sources

The simulation results in the preceding subsection show that the proposed protocol realizes timely and lossless medium access for LDS packets in mobile ad hoc networks. This subsection shows how the proposed protocol performs in supporting multiple levels of priority for LDS messages. We did not further investigate the performance of IEEE 802.11 MAC protocols because of their problematic performance shown in the preceding subsection for supporting LDS packets.

In our simulations testing the competition among LDS message sources, five sources competed with each other at locations shown in Fig. 8 and each LDS source had a distance of 200 meters to source 2. All the five sources needed to disseminate a burst of LDS packets at the same time in our simulations.

We first show the simulation results for the case in which LDS packets have a single level of priority in accessing the medium (i.e., all of them have the same priority over normal packets). The simulation results are shown in Fig. 9 as an event map for the competing sources. In the figure, the horizontal axis shows the time, while the vertical axis shows the identification number of the node at which an event happens. There are three types of events shown in the figure, which are successful medium access (i.e., backoff timer expires successfully), transmission interruption, and successful packet delivery. The three types of events are denoted by triangles, crosses, and rings, respectively, in Fig. 9.

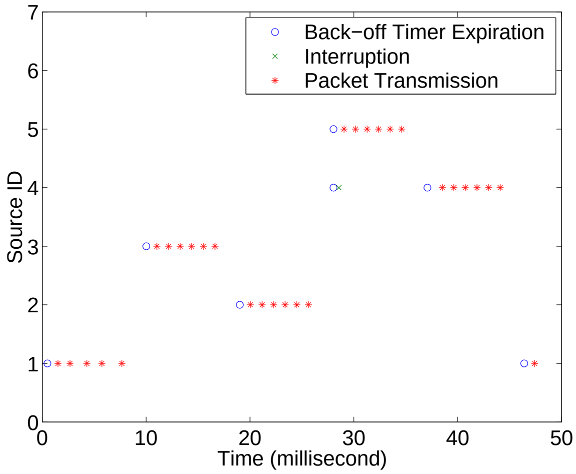

Figure 9: The Event Map for The Competing LDS Message Sources. The Five LDS Message Sources Have the Same Level of Priority of Medium Access.

As shown in Fig. 9, the five sources successfully deliver their LDS packets in sequence. Source 2, 3, and 5 have a single successful access to the medium before delivering their packets. However, both sources 4 and 1 successfully access the medium twice before delivering their packets. Source 4 has a collision with source 5 at about the 28th ms. The collision is, however, resolved immediately due to the pulses in the control channel. Source 4 therefore accesses the medium twice. Source 1 also accesses the medium twice before sending out its LDS packets, but without collision. Source 1, in fact, releases the medium voluntarily after sending out its fifth LDS packet, which indicates that the last LDS packet in its burst does not follow the other packets closely.

In the above case of same-level priority for all LDS message sources, source 4 is the last to use the medium among the five sources, while source 5 is the next to the last. It is possible, however, that the LDS messages of these two sources have much stricter requirements on delay. In such a case, node 4 and node 5 are unduly delayed for 30 to 40 ms in accessing the medium because of the competition from the other LDS message sources that have lower requirements on delay. To demonstrate how the proposed PreempPrio-MAC protocol can help in such a case, we assigned source 4 the highest level of priority, which is level 3. Meanwhile, source 5 was assigned

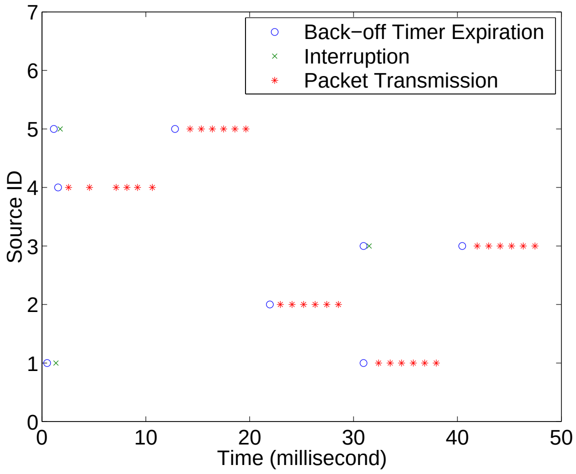

Figure 10: The Event Map for Competing LDS Message Sources. The Five LDS Message Sources Do Not Have the Same Level of Priority in Accessing the Medium. Source 4 Has a Priority Level of 3 (The Highest Level), Source 5 Has a Priority Level of 2, And Other Sources Have a Priority Level of 1.

a priority level of 2.

The simulation results for this new case are shown in Fig. 10. As shown in Fig. 10, source 1 successfully accesses the medium first because its LDS packets arrive at its MAC sub-layer first. However, before it finishes transmitting its first LDS packet, it is interrupted by source 5, which has higher priority than source 1. Similarly and again, source 5 is interrupted by source 4, which has the highest priority among all the five sources. Source 4 successfully finishes transmitting its LDS packets after interrupting source 5. Source 5, which has the second highest priority, then disseminates its LDS packets. After source 4 and source 5 finish broadcasting their LDS packets, other sources disseminate their LDS packets in sequence, although source 1 and source 3 have an immediately-resolved collision at about the 31st ms.

4 Summary

Existing MAC protocols for wireless packet networks do not pay much attention on broadcast packets, which leaves an open problem of a MAC protocol to support applications that rely on the dissemination of loss and delay sensitive (LDS) messages in wireless networks. This paper presents and evaluates a new MAC protocol that is designed to support LDS message dissemination in wireless packet networks. The proposed protocol solves the hidden terminal problem for LDS messages and meanwhile provides multiple levels of strict priority for them in medium access. Extensive simulations showed the ideal performance of the proposed protocol.

References

[1] J. Peng, "Timely and lossless medium access in ad hoc networks," in *The $39th$ Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, March 2005.

[2] IEEE 802.11 wireless local area networks. [Online]. Available: http://grouper.ieee.org/groups/802/11/

[3] United states department of transportation - intelligent transportation systems. [Online]. Available: http://www.its.dot.gov/

[4] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Transactions on Communications*, vol. 23, pp. 1417–1433, 1975.

[5] P. Karn, "MACA - a new channel access method for packet radio," in *Proceedings of the 9th ARRL Computer Networking Conference*, Ontario, Canada, 1990.

[6] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a medium access protocol for wireless LANs," in *Proceedings of the ACM SIGCOMM Conference*, London, United Kingdom, August 1994.

[7] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *Proceedings of the ACM SIGCOMM Conference*, September 1995.

[8] J. Deng and R. Chang, "A priority scheme for IEEE 802.11 DCF access method," *IEICE Trans. Commun.*, vol. E82-B, pp. 96–102, 1999.

[9] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *Proceedings of the IEEE INFOCOM Conference*, Anchorage, Alaska, April 2001.

[10] M. Barry, A. T. Campbell, and A. Veres, "Distributed control algorithms for service differentiation in wireless packet networks," in *Proceedings of the IEEE INFOCOM Conference*, Anchorage, Alaska, April 2001.

[11] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed priority scheduling and medium access in ad hoc networks," *ACM Wireless Networks*, vol. 8, pp. 455–466, November 2002.

[12] X. Yang and N. H. Vaidya, "Priority scheduling in wireless ad hoc networks," in *Proceedings of the ACM MOBIHOC Symposium*, EPFL Lausanne, Switzerland, June 2002.

[13] J.L.Sobrinho and A.S.Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," *Bell Labs Technical Journal*, pp. 172–187, 1996.

[14] C. Wu and V. O. K. Li, "Receiver-initiated busy-tone multiple access in packet radio networks," in *Proceedings of the ACM SIGCOMM Conference*, Stowe, Vermont, August 1987.

[15] J. Deng and Z. J. Haas, "Dual Busy Tone Multiple Access (DBTMA) a multiple access control scheme for ad hoc networks," *IEEE Transactions on Communications*, vol. 50, pp. 975–985, June 2002.

[16] The network simulator - ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/