Priority Scheduling in Wireless Ad Hoc Networks

(Draft Version)

Xue Yang and Nitin.H.Vaidya
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Abstract— **Ad hoc networks formed without the aid of any established infrastructure are typically multi-hop networks. Location dependent contention and "hidden terminal" problem make priority scheduling in multi-hop networks significantly different from that in wireless LANs. Most of the prior work related to priority scheduling addresses issues in wireless LANs. In this paper, priority scheduling in multi-hop networks is discussed. We propose a scheme using two narrow-band busy tone signals to ensure medium access for high priority source stations. The simulation results demonstrate the effectiveness of the proposed scheme.**

Keywords—**Ad hoc network, medium access control, priority scheduling, busy tone.**

I. Introduction

WITH advances in wireless communications and the growth of real-time applications, wireless networks that support quality of service (QoS) have recently drawn a lot of attention. In order to provide differentiated service to real-time and non-real-time packets, the medium access control protocol must provide certain mechanisms to incorporate differentiated priority scheduling, such that higher priority traffic can be transmitted in preference to lower priority traffic.

There are two standards for wireless networks that cover MAC sub-layer: the European Telecommunications Standards Institute (ETSI) High Performance European Radio LAN (HIPERLAN) [1] and the IEEE 802.11 WLAN [2]. HIPERLAN explicitly supports QoS for packet delivery in wireless LANs. IEEE 802.11 may carry traffic with time-bounded requirements using PCF (Point Coordination Function), which needs the coordination of an "Access Point". Neither of them can provide effective priority scheduling in ad hoc networks.

By wireless LAN, we mean a network in which all stations[1] are within each other's transmission range. On the other hand, in multi-hop networks, two stations that cannot hear each other may still compete with each other for the channel due to the "hidden terminal" problem. In such environments, in addition to local channel information, the channel status near the neighboring nodes also has to be considered to ensure priority scheduling.

Another difference between Wireless LANs and multi-hop networks with respect to priority scheduling is that different flows in multi-hop networks have different degree of contention. Here, we define the contention degree for a flow as the number of flows with which it is competing for the channel. While each flow competes for the channel with all other flows in wireless LANs, in multi-hop networks, different flows may experience different situations depending on the network topology and flow pattern. For example, in a multi-hop network, it is possible that flow A has contention degree of 10 while flow B just competes with flow A. Under such a circumstance, it might be easier for flow B to access the channel.

While there is some research related to priority scheduling in wireless networks [3] [4] [5] [6] [7] [8] [9] [10], most of these schemes can only work well in a wireless LAN. In this paper, we propose a scheme using two narrow-band busy tone signals to achieve effective priority scheduling in ad hoc networks.

The rest of this paper is organized as follows. Section II presents the related work. The problem of priority scheduling in multi-hop networks is discussed in section III. Section IV describes the proposed busy tone priority scheduling (**BTPS**) protocol. Performance evaluation is presented in section V. Finally, we present our conclusions in section VI.

II. Related Work

Medium Access Control (MAC) protocols that aim to provide differentiated services should be able to meet requirements of traffic with different priority classes. If a high priority flow's traffic pattern satisfies the behavior described in the service agreement, its packets should be delivered in preference to other packets with lower priorities.

This research is supported in part by National Science Foundation grant ANI-9973152.

[1] We use the terms *station* and *node* interchangeably

On the other hand, flows with lower priorities should use as much bandwidth as possible after the transmission requirements of higher priority flows have been satisfied.

In general, there are two directions in wireless MAC protocols to facilitate channel access privilege of high priority traffic: reservation based schemes and contention based schemes.

Reservation based schemes usually make some assumptions about high priority traffic. For example, high priority traffic is assumed to be periodic with fixed arrival rate. For reservation based schemes, when resources are reserved but unused, they are often wasted. A typical example of a reservation based MAC protocol is GAMA/PS [3]. GAMA-PS divides time into a sequence of cycles; each cycle begins with a contention period and ends with a "group-transmission" period. The group-transmission period is divided into a set of zero or more individual transmission periods, each for a station in the "transmission group". A station with data to send competes for membership in the "transmission group" during the contention period; also, by listening to the channel, a group member becomes aware of how many stations are in the group and of its own position within the group. In this case, members of the transmission group take turn transmitting data, and collision is avoided. However, a basic requirement for this protocol is that each station can hear the transmissions of other stations, which limits the use of the protocol to wireless LANs.

The MACA/PR protocol [6] extends the reservation based scheme to multi-hop networks. The first data packet of a high priority flow makes reservations along the route to the destination. Each station maintains a reservation table (RT) which keeps track of the transmitting and receiving "reserved windows" for neighbors within a two-hop neighborhood. Low priority sources are only allowed to fill in empty windows. In order for the reservation scheme to work, the size of high priority packets must be prespecified for each connection, and the size of low priority packets must be bounded so as not to interfere with the reservation constraints.

Unlike the reservation based schemes, contention based schemes are probabilistic. Flow scheduling decision is made locally, and contention is resolved probabilistically. As an example, reference [4] uses "black burst" to help high priority flows contend for the channel. After channel becomes idle, a high priority flow has shorter waiting time before it transmits the "black burst", other low priority flows which have longer waiting time will drop out of contention once they hear the "black burst" during their waiting time. This scheme thus provides a way for the high priority source stations in a wireless LAN to reserve the channel by occupying the channel with "black burst". Reference [5] further generalizes this scheme to "ad hoc carrier sense multiple access wireless network", which is defined as a wireless network without hidden nodes. That is, each source station in such a network can always sense the possible interfering transmissions. However, this is not the case in most ad hoc networks. More often, "hidden terminals" do exist in ad hoc networks, and nodes cannot always sense each other's transmissions. Thus, the scheme in [5] cannot be applied to general ad hoc networks.

Several researchers propose some simple modifications to the IEEE 802.11 Distributed Coordination Function (DCF) to incorporate differentiated service. IEEE 802.11 DCF defines a collision avoidance mechanism to resolve contention among different stations willing to access the medium. Each station chooses a random number between zero and a given "Contention Window" as the backoff duration. After sensing the channel to be idle for a suitable "interframe space" duration, each station waits until the backoff timer has been counted down to zero before accessing the channel. A station freezes its backoff timer if it senses a busy channel, and then continues to count down the backoff timer when the channel becomes idle for "interframe space" duration again. If collisions occur, the colliding stations will exponentially increase their "Contention Window" by a factor of 2. The value of "Contention Window" is constrained to be between CW_{min} and CW_{max}. A source station sends a "RTS" (request to send) first. If it gets a "CTS" (clear to send) back from the receiver, the data packet will be sent, followed by an "ACK" from the receiver. In the case that a "RTS" is not followed by a "CTS", or "Data" is not followed by an "ACK", collision is assumed to have occurred.

Summarizing, there are two "waiting stages" in IEEE 802.11 before the station accesses the channel.

- The "interframe space"(IFS) stage.
- The backoff stage, whose duration is a random value between zero and the "Contention Window".

In [7], [8], [9] and [10], various schemes have been proposed to modify the backoff stage so that different priority source stations use different "Contention Window" generation functions. For example, [7] proposes that high priority source stations randomly choose the backoff interval from $[0, 2^{i+1} - 1]$ and low priority source stations choose from $[2^{i+1}, 2^{i+2} - 1]$, where i is the number of consecutive times a station attempts to send a packet. [9] proposes to set different values of CW_{min} and CW_{max} for different priority classes. [8] proposes that instead of using the exponential factor of 2 after a collision, different priority classes use different exponential increase factor. Stations with lower priority increase their "Contention Window" much faster than the stations with higher priority. One drawback faced by [7], [9] and [8] is that high priority flows may possibly experience more collisions compared to their low priority counterparts in multi-hop networks. As a result, "high priority" flows cannot be ensured to have smaller "Contention Window", hence, the priority of channel access cannot be ensured either. In order to adapt better to multi-hop networks, in [10], a packet's priority information is piggybacked in the RTS/CTS/Data/ACK frames. Based on overheard packets, each station maintains a scheduling table, which records priority information of flows that are within two-hop neighborhood. The backoff duration is generated based on the scheduling table. However, this scheme suffers from incomplete scheduling table which is caused by collisions, location dependent errors, node mobility and partially overlapping transmission regions.

All of the above schemes, which propose to modify the backoff interval of IEEE 802.11 to incorporate differentiated service, suffer from one major drawback as described below:

- As the backoff timer for a low priority packet is frozen only when the channel becomes busy, it will **continue** to count down each time when the channel becomes idle again. Thus, eventually a low priority packet that arrived earlier might have the shortest backoff interval. In such cases, "priority reversal" occurs in that the low priority packet has a shorter backoff interval than backlogged high priority packets, and grabs the channel.

An example is illustrated in Fig. 1.

Suppose that two ranges of backoff interval [0, 15] and [16, 31] are now used respectively by high priority and low priority packets. Nodes 1 and 4 have high priority packets (flows 1, 2) to node 2 while nodes 3 and 5 have low priority packets for node 2 (flows 3, 4). At time t1, nodes 1, 3 and 5 had packets backlogged with backoff intervals 10, 17 and 18 respectively. Node 1 began its transmission at time t2, so nodes 3 and 5 froze their backoff counters with the remaining values of 7 and 8. During node 1's transmission, at time t3, a high priority packet arrived at node 4. Node 4 chose 9 slots as the backoff interval. When node 1 finished its transmission at time t4, nodes 4, 3 and 5 began to count down backoff interval after "interframe space" duration. Hence, node 4, the high priority source node, had the largest backoff counter. Consequently, node 4 lost the channel access to nodes 3 and 5.

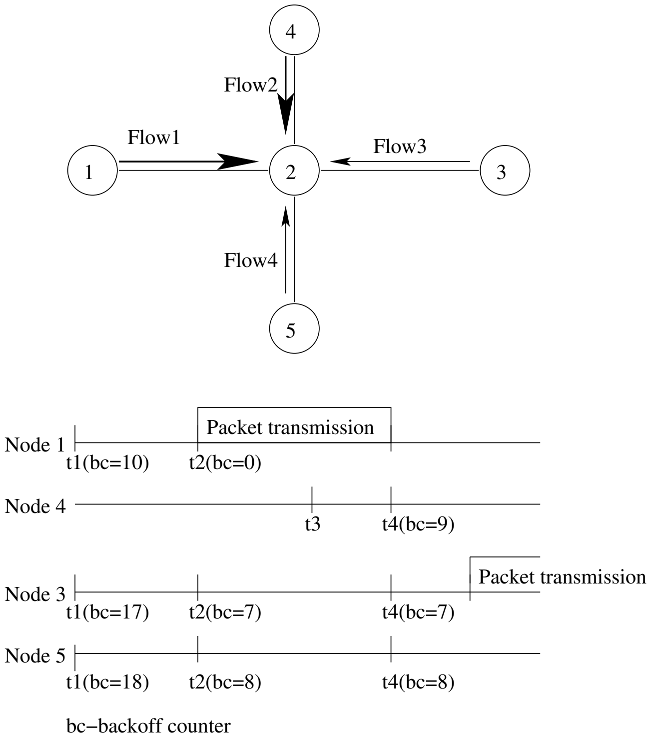

Fig. 1. Priority reversal.

As we mentioned earlier, IEEE 802.11 requires each station to wait for the channel to be idle for "interframe space(IFS)" duration before counting down the backoff interval. IEEE 802.11 defines 4 types of IFS, which are used to provide different priorities for different transmissions. Packets with shorter IFS have higher priority. SIFS is the minimum interframe space, which is used to separate transmissions belonging to a single dialog, i.e., CTS, DATA and ACK transmissions, thus giving them highest priority. PIFS is used by the PCF (Point Coordination Function) to give the Access Point higher priority over other stations. DIFS is used by a station willing to start a new transmission. EIFS is the longest IFS used by a station that has received a packet that it could not understand; this is needed to prevent the station from colliding with a future packet belonging to an on-going dialog.

Unlike IEEE 802.11 Distributed Coordination Function (DCF) in which all new transmissions use DIFS as the "interframe space(IFS)", [7] and [8] propose that different priority source stations can apply different IFS. Specifically, one of the schemes proposed in [8] works in the following way. Assume that there are two priority classes: one is high priority and the other is low priority. Then, the IFS of a low priority flow is defined as the sum of IFS and maximum contention window of high priority flows. The high priority packets are constrained to increase their contention window no larger than the above maximum value. That is, let LIFS represent the IFS for low priority flows, HIFS represent the IFS for high priority flows, and Cwh represent the maximum contention window of high priority flows. Then we have:

$$LIFS = HIFS + Cwh$$

This scheme sacrifices available network capacity to ensure the transmissions of high priority flows. Since the entire IFS duration is enforced before each station can continue to count down the backoff interval, this scheme avoids the "priority reversal" problem mentioned earlier. However, there is a critical trade-off between making full use of bandwidth and ensuring priority. If the maximum contention window of high priority flows is constrained to be too small, they will experience high degree of contention. On the other hand, if this parameter is chosen to be too large, significant bandwidth will be wasted by making low priority flows wait very long unnecessarily when high priority flows are not backlogged.

Among several choices of modifying IEEE 802.11 DCF, [8] shows that the scheme using different IFS for different priority classes, as described above, works best. For this reason, this scheme is chosen to be the one that we compare our scheme's performance with. Considering only two priority classes, this scheme is implemented in the following way: high priority flows use DIFS as the IFS. The sum of DIFS and Cwh (as defined above) is used as the IFS of low priority flows. Throughout the rest of this paper, this scheme is called **"PMAC"**(Priority MAC) for convenience.

III. Priority Scheduling in Multi-hop Networks

We consider two priority classes: high priority and low priority.

A. Impact of "Hidden Terminals" on Priority Scheduling

Fig. 2. Impact of "hidden terminals" on priority scheduling

Consider a very simple three-hop scenario in Fig. 2. Node 0 has high priority packets for node 1 (flow 1) and node 2 has low priority packets for node 3 (flow 2). Flow 1 and flow 2 conflict with each other since node 2's transmission will interfere with node 1's reception of any other packets. When both flows are backlogged, how to ensure the channel access priority of flow 1?

The scheme proposed in [8], which we refer to as "PMAC" in section II, tries to solve this problem by forcing node 2 to wait for a longer IFS after the channel becomes idle. However, as we mentioned earlier, there is a critical trade-off between making full use of bandwidth and ensuring priority.

The key point here is that, when node 0 has a high priority packet backlogged, node 2 should be aware of that and defer its transmission; on the other hand, if node 0 is not backlogged, node 2 should maximize its own throughput. This objective can be achieved by using two narrow-band busy tone signals (BT1 and BT2) as proposed in this paper. The basic idea (as elaborated later) is that whenever a high priority packet is backlogged at node 0, it will send a BT1 every M slots before it acquires the channel, where M is a parameter of the proposed scheme. In Fig. 2, when node 1 hears this BT1, it will send a BT2. All nodes with low priority packets that hear either BT1 or BT2 will defer their transmissions for some duration. In this way, channel access priority of node 0 can be ensured. Certainly, if there is no high priority packet backlogged at node 0, node 2 will not hear any busy tone signal, hence, its channel access will not be affected at all. The details of this protocol are described in Section IV.

B. Impact of Collisions on Priority Scheduling

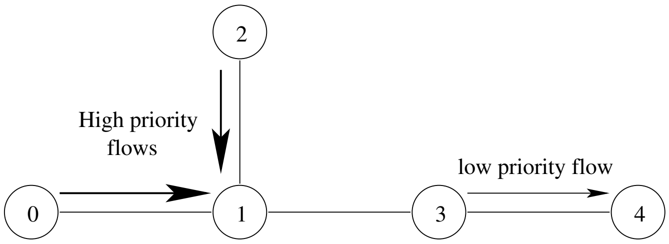

Fig. 3. Impact of collisions on priority scheduling

In Fig. 3, nodes 0 and 2 have high priority packets for node 1 while there is a low priority flow from node 3 to node 4. When node 3 transmits to node 4, node 1 cannot receive any packet from node 0 or 2 during that transmission. Now suppose the transmissions of nodes 0 and 2 collide at node 1 (this can occur with non-negligible frequency). The time period in which nodes 0 and 2 detect the collision and resolve the channel contention could be long. Unless node 3 defers its transmission during this entire period, nodes 0 and 2 are likely to lose the channel access to node 3. However, how can node 3 know that collision occurred between high priority nodes 0 and 2? Similarly, how can node 3 know that the contention between nodes 0 and 2 has been resolved and both of them have finished the transmissions of backlogged high priority packets?

In multi-hop networks, under severe contention amongst high priority flows it is a challenge to ensure their priority over low priority flows. The major difficulty is that every node can only sense its local channel status. In the example above, even if nodes 0 and 2 are experiencing continuous collisions, node 3 still may sense its channel as free and start its transmission.

The scheme proposed in this paper solves this problem as follows. During the procedure of channel access of nodes 0 and 2, they will send BT1 signal every M slots until the packet is sent on the data channel, where M is a parameter to be set as mentioned earlier. Node 1 will send BT2 after sensing BT1. If the transmissions of nodes 0 and 2 collide at node 1, they will detect the collision after some duration, which is called "CTS-Timeout" in the case of IEEE 802.11 DCF using RTS/CTS handshake. After the collision is detected, the channel access procedure will start once again, during which BT1 and BT2 will again be sent periodically. We require low priority source nodes that sense BT1 or BT2 signal to defer their transmissions for the "CTS-Timeout" duration. This ensures channel access of high priority packets as elaborated in Section IV.

IV. Proposed Busy Tone Priority Scheduling (BTPS) Scheme

The scheme proposed in this paper is a contention based protocol. The proposed scheme makes use of two busy tone signals, and borrows some mechanisms from IEEE 802.11 DCF. The proposed protocol is called "Busy Tone Priority Scheduling(**BTPS**)". We now describe the protocol, followed by an example in Fig. 5.

A. *Channel Requirement*

In the proposed BTPS scheme, two narrow-band busy tone signals named BT1 and BT2 are used. Reference [11] previously proposed the use of two busy tone signals to provide higher network utilization. The work in [11] has a different objective and different mechanism compared to the priority scheduling protocol proposed in this paper.

Low priority source stations determine the presence of high priority packets by sensing the carrier on the busy-tone channel. According to [12], the time period of 5μs is sufficient for the busy tone signal to be detected if 1% of total channel frequency spectrum[2] is assigned to each busy tone channel (including guard band). To ensure adequate spectral separation between two busy tone channels, they can be put at the two ends of the channel spectrum as Fig. 4 shows. Now, the total available channel bandwidth is divided into three parts: BT1 channel, Data channel and BT2 channel, with respective bandwidth percentage of 1%, 98%, and 1%. The resulting data channel has a bit rate of 1.96 Mbps.

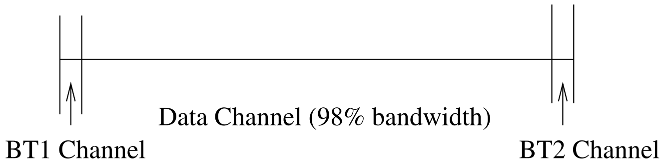

Fig. 4. Channel Spectrum Division

In general, it is hard to require a node to have the capability of receiving while it is transmitting, or transmitting to more than one channel at the same time. The proposed BTPS protocol only requires that stations be able to monitor the carrier status of the data channel as well as two busy tone channels while the station is idle and lock onto the signal on the data channel as desired. Here, a station is defined to be idle when it is not transmitting to any channel, and it is not receiving a packet from the data channel. Since we only need to detect the existence of busy tone without decoding, it should not be difficult for a station to have such capabilities. Once stations begin to receive from or transmit to the data channel, the status of busy tone channel can be ignored. The busy tone channel's sensing threshold is set the same as data channel's sensing threshold.

B. *Channel Access Procedure with the use of dual busy tone*

In BTPS, busy tone serves as the indication of backlogged high priority packets. All packets are transmitted on the data channel. Each dialog begins with RTS/CTS handshake, followed by the transmissions of Data/ACK packets.

As in IEEE 802.11, each station, before accessing the channel, needs to wait for the channel to be idle for the period of "interframe space (IFS)", then enter the backoff stage. The length of backoff interval is randomly chosen between zero and the value of "Contention Window". When collision occurs, the "Contention Window" will be exponentially increased by the factor of 2. Stations will freeze their backoff timers once they sense data channel is busy. At the end of backoff stage, stations are allowed to acquire the channel. Time is slotted and each unit is called one Slot-Time.

The difference between IEEE 802.11 Distributed Coordination Function (DCF) and BTPS is that high priority and low priority source stations behave differently during "IFS" and "backoff" stages in BTPS.

• High Priority Source Stations: The DIFS is used as the interframe space for high priority source stations. Dur-

[2]The channel frequency spectrum is 22 MHz with IEEE 802.11 DSSS [2].

ing DIFS and backoff stages, the high priority source stations send a BT1 pulse (5μs duration) every M slots. Between two consecutive busy tone pulse transmissions, there should be at least one Slot-Time interval so that these stations have a chance to listen to data channel. Therefore, M could be any value that is larger than 2, depending on the choice of IFS for low priority source stations. The principle is that the IFS of low priority stations should be larger than M slots, so that they can always sense the busy tones before they attempt to acquire the channel. In our implementation, M is set to 3.

- Stations that sense BT1: High priority source stations will disregard BT1. Any other station that senses a BT1 will send a BT2 pulse (5μs duration) if it is not receiving a packet from the data channel. It will also defer its transmission of a low priority packet. Specifically, RTS for a low priority packet is deferred for "CTS-Timeout" duration after receipt of a BT1. Special attention also needs to be paid to the transmission interval of BT2. Between two consecutive BT2 pulses, there should be at least one Slot-Time interval to make sure that the stations, which transmit BT2 after sensing BT1, have a chance to receive packet from data channel. That is, a station will send BT2 pulse at most once every two slots.
- Stations that sense BT2: High priority source stations will disregard BT2. Any other station that senses a BT2 will defer its RTS for a low priority packet for "CTS-Timeout" duration.
- Low priority source stations: DIFS plus one Slot-Time is used as the "interframe space" for low priority source stations. In the case of IEEE 802.11 DSSS [2], DIFS lasts for two and half Slot-Time. Since busy tone will be initiated every three slots by high priority stations, low priority source nodes that wait for at least three and half Slot-Time will sense the busy tone and defer their transmissions.

C. Occupancy of Data Channel using black burst

During the channel access procedure described above, a station may transmit BT1 or BT2. However, the same station could be the receiver of a high priority packet for which an RTS may be transmitted while it is sending BT1 or BT2. Since a station cannot receive while it is transmitting, the high priority packet intended for this station will be missed during its busy tone transmission. The scenario in Fig. 3 can be used to illustrate the situation. After sensing BT1 from node 0, node 1 will send BT2 correspondingly. But when node 1 is transmitting BT2, node 2, another high priority source node, could possibly be sending RTS to node 1 on the data channel. Without taking care of such a situation, node 1 will miss the high priority packet from node 2.

Taking into account several factors including data channel carrier detection time, turnaround time of stations from receiving mode to transmitting mode as well as the transmission time of BT2, BTPS requires that each high priority source station send a two slot duration of "black burst" before the transmission of RTS packet on data channel. The "black burst" is used to occupy the channel. With "black burst" ahead of useful data, the receivers will either detect that data channel is busy before turning to transmit busy tone, or be able to correctly receive packet from data channel after the transmission of a busy tone.

Now, back to our example. After the transmission of BT2, node 1 will sense the carrier on data channel and begin to receive the signal. Because of the two slot duration of "black burst" ahead of RTS packet, node 1 can still receive the RTS packet correctly from node 2.

There is no need to add "black burst" before CTS, Data or ACK packets.

D. Summary of BTPS protocol

The behavior of the BTPS protocol is summarized in Fig. 5. The high priority source station in Fig. 5(a) will send BT1 every 3 slots during DIFS and backoff stage. Once the backoff counter is counted down to zero and the channel is idle, a "black burst" which lasts two Slot-Time long will be sent first, followed by a RTS packet. After getting a CTS reply, the data packet will be sent, followed by the reception of an ACK. From the point of sending "black burst" to the time of receiving the ACK, no busy tone signal will be transmitted. Any other station that senses BT1, as shown in Fig. 5(b), will transmit BT2 provided that it is not receiving from data channel. Each time when stations sense busy tone (BT1 or BT2), the transmissions of low priority packets will be deferred for "CTS-Timeout" duration, which is shown in Fig. 5(b) and Fig. 5(c).

V. Performance Evaluation

In this section, simulation results are presented to demonstrate the effectiveness of the proposed BTPS protocol. The simulation results for PMAC [8] are also shown for comparison. Recall that PMAC is a modified version of IEEE 802.11 DCF, which chooses IFS for low and high priority flows differently to attempt to achieve priority scheduling. As we mentioned earlier, Cwh is a critical parameter for PMAC, and it is difficult to adapt this parameter to dynamic network situations. However, in the simulation, being aware of the number of high priority flows and traffic load, we try to choose a suitable value to demonstrate a reasonable performance for PMAC. In some scenarios, results of IEEE 802.11 DCF are also presented to show the baseline. The performance metrics we use in-

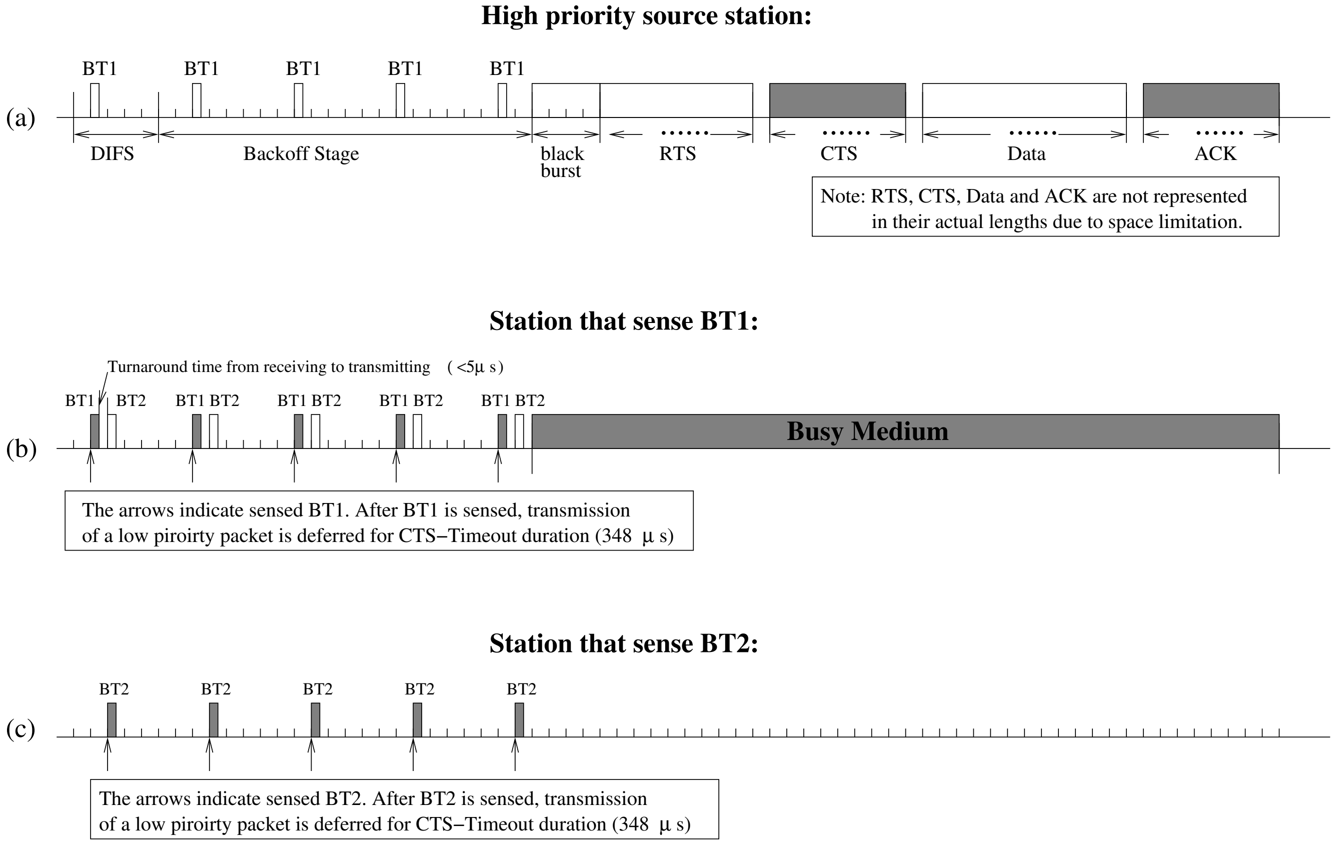

Fig. 5. Behavior of BTPS protocol. One Slot-time is 20μs. The duration between consecutive ticks, shown as short bars in the figure, is 10μs. Black boxes represent received signal, and white boxes represent transmitted signal. Figure (a) shows the behavior of a high priority source station which has a packet backlogged. Figure (b) shows behavior of stations that sense BT1, while figure (c) shows behavior of stations that sense BT2.

clude "Delivery Ratio of High Priority Packets", which is the ratio of high priority flows' throughput over their sending rate; and "Aggregate Throughput", which is the aggregate throughput of all high and low priority flows. For PMAC [8], a higher value of Cwh improves the first metric; but degrades aggregate throughput, and vice versa. Our scheme can improve on PMAC with respect to both metrics.

A. Simulation Model

All the simulation results are based on a modified version of ns-2 network simulator from USC/ISI/LBNL [13], with wireless extensions from the CMU Monarch project [14]. The extensions provide a wireless protocol stack including IEEE 802.11. The radio interface model approximates the first generation WaveLan radio interface with 2 Mbps bit rate and 250 meter transmission range using omnidirectional antenna. The traffic sources are chosen to be constant bit rate (CBR) sources using packet size of 512 bytes. Cwh for PMAC is set to 32 slots.

Since our objective is to demonstrate MAC protocol's performance to deliver high priority packets, mobile situations are not simulated here. However, the behavior of BTPS protocol itself will not be impacted by mobility.

B. Scenario 1

In this scenario, 24 nodes are arranged in a 4×6 grid with a grid spacing of 200 meters. The flow pattern is as shown in Fig. 6. Fig. 7 plots the flows' conflict graph. The conflict graph is defined as G=(V, E), in which V is the set of all flows, and an edge (f_i, f_j) belongs to E if and only if flows f_i and f_j conflict with each other (i.e., they cannot transmit simultaneously). Among all flows, flows 5 and 8 have the highest contention degree, while flows 1, 3, 10, 12 have the lowest contention degree.

The number of high priority flows is increased from 0 to 6 in our simulations. The corresponding high priority flows for each case are given in Table I. The traffic sending rate for each high priority flow in each case is 180 Kbps, while all remaining low priority flows have aggressive sending rate of 1500 Kbps.

Fig. 8 plots the delivery ratio of high priority packets

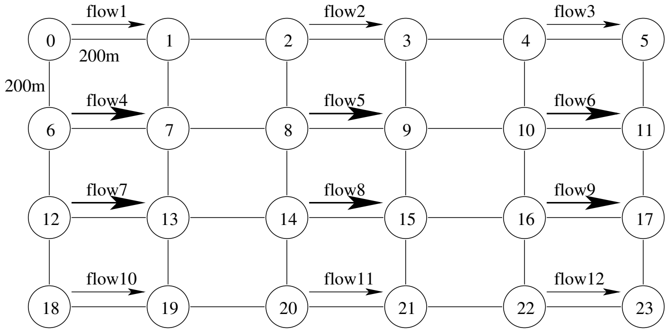

Fig. 6. Network topology of scenario 1

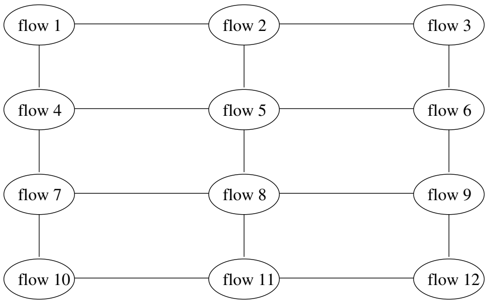

Fig. 7. Conflict graph for flows in scenario 1

versus the number of high priority flows. The proposed BTPS protocol can deliver almost all high priority packets in each case, while the delivery ratio of PMAC [8] begins to drop when the number of high priority flows is 3. When there are six high priority flows, the performance gap between BTPS and PMAC in terms of high priority packets' delivery ratio reaches 14%. Because IEEE 802.11 DCF does not provide differentiated service and the high priority flows simulated have higher contention degree, IEEE 802.11 delivers very few high priority packets.

Fig. 9 presents the aggregate throughput for BTPS, PMAC and IEEE 802.11. When all flows are low priority (i.e., number of high priority flows is 0), PMAC only achieves 83.5% aggregate throughput compared to that of IEEE 802.11. In PMAC, for each packet's transmission, the waiting time of low priority source nodes in "interframe space" stage is 32 (Cwh) slots more than the corresponding waiting time in IEEE 802.11. This causes the 16.5% loss of aggregate throughput. Furthermore, the larger the value of Cwh, the more is the loss in aggregate throughput. If we reduce the value of Cwh, the throughput loss can be smaller, but the deliver ratio for high priority packets would be worse. On the other hand, BTPS achieves 97.4% aggregate throughput of IEEE 802.11. The loss of throughput is mainly caused by the 2% bandwidth given to busy tone channels in BTPS.

When there are high priority flows, IEEE 802.11 schedules a different set of flows compared to priority scheduling protocols BTPS and PMAC, hence achieves much more throughput at the cost of starving high priority flows. The situation is elaborated below.

Number of high priority flows	High priority flow ID
0	
1	flow 4
2	flow 4, 6
3	flow 4, 5, 6
4	flow 4, 5, 6, 8
5	flow 4, 5, 6, 7, 8
6	flow 4, 5, 6, 7, 8, 9

TABLE I
THE HIGH PRIORITY FLOWS IN SCENARIO 1

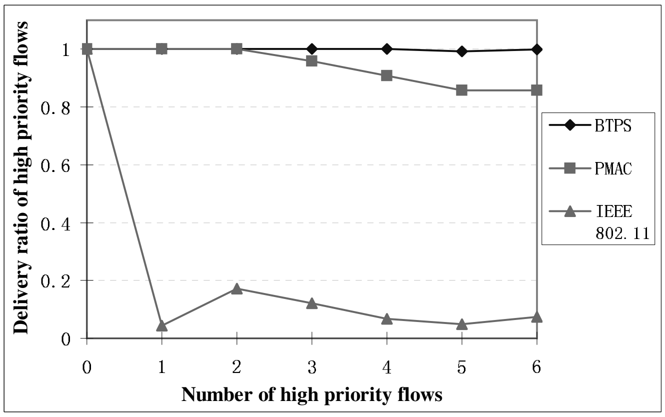

Fig. 8. Delivery ratio of high priority packets, comparison between BTPS, PMAC and IEEE 802.11

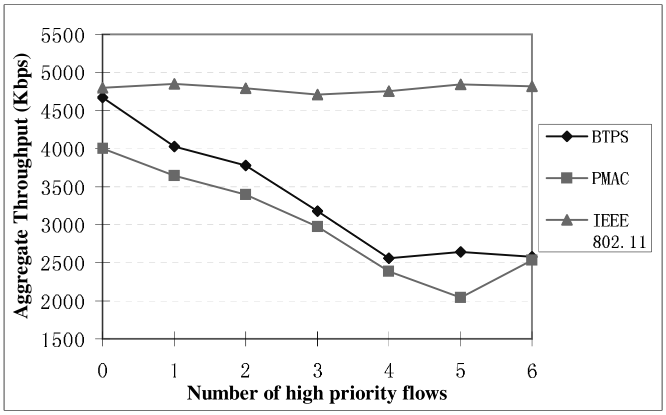

Fig. 9. Aggregate throughput comparison between BTPS, PMAC and IEEE 802.11

For the scenario with six high priority flows, we plot each individual flow's throughput for BTPS, PMAC and IEEE 802.11 DCF in Fig. 10. The highest throughput in this situation can be achieved by scheduling flows 1, 3, 10, and 12 at all times since they have the lowest contention degree and most aggressive sending rate. However, this maximum throughput is achieved at the cost of starving other flows, particularly, the high priority flows 4, 5, 6, 7, 8 and 9. From the results shown in Fig. 10, IEEE 802.11 DCF performs in this way and achieves the highest aggregate throughput of 4818 Kbps[3]. On the other hand, BTPS and PMAC give channel access preference to high priority flows, but at the cost of decreased aggregate throughput. BTPS achieves aggregate throughput of 2579 Kbps and PMAC achieves 2539 Kbps. The reason why both BTPS and PMAC lose throughput in comparison with IEEE 802.11 is because high priority flows have higher contention degree in the simulated scenario. As shown in Fig. 7, for example, when flow 4 is transmitting on the data channel, flows 1, 5, 7 cannot be scheduled. Similarly, when flow 5 is transmitting, flows 2, 4, 6, 8 cannot use the channel either.

Note the two high priority flows (flows 5 and 8) with the highest contention degree. PMAC just delivers 80% of packets for flow 5, and 34% for flow 8, while BTPS delivers almost all high priority packets for all flows. PMAC is unable to deliver many high priority packets due to contention among the high priority flows. With PMAC, the problem illustrated in section III.B occurs often, resulting in low priority traffic gaining channel access instead of high priority traffic. Thus, PMAC [8] delivers more low priority packets from flows 1 and 10 but fewer high priority packets from flows 5 and 8 than the proposed BTPS protocol. Since flows 1 and 10 do not experience much contention, PMAC is able to achieve high throughput for these flows (at the cost of low throughput of high priority flows 5 and 8). This explains why the difference of aggregate throughput between PMAC and BTPS is small when the number of high priority flows is six.

C. Scenario 2: Random Topology

We generate eight random topologies in a 1000m$\times$1000m area. The total number of nodes in this area is increased from 10 to 80 with a step size of 10, and flows are randomly chosen between two nodes which are one hop away. Among all flows, half are high priority flows with sending rate of 120 Kbps, the remaining low priority flows have aggressive sending rate of 1500 Kbps. Table II shows the number of high priority flows for each simulated topology.

[3]Recall that our proposed scheme can achieve the aggregate throughput comparable to IEEE 802.11 when there are no high priority flows.

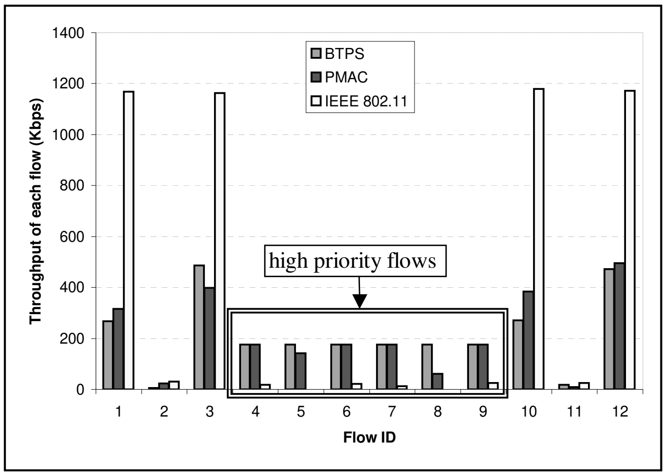

Fig. 10. Throughput of each flow in scenario 1 with six high priority flows

Num. of nodes	Total num. of flows	Num. of high priority flows
10	7	4
20	14	7
30	24	12
40	33	17
50	43	22
60	53	27
70	65	33
80	73	37

TABLE II
THE NUMBER OF HIGH PRIORITY FLOWS IN RANDOM TOPOLOGIES

The delivery ratio of high priority packets is shown in Fig. 11, from which we can see that BTPS delivers more high priority packets than PMAC in most cases. Only when there are only 10 or 20 nodes and the corresponding numbers of high priority flows are 4 or 7 respectively, does PMAC deliver as many high priority packets as BTPS. In the case of 80 nodes, the delivery ratio difference between BTPS and PMAC reaches 22.5%. The simulation results demonstrate that severe contention among high priority flows will cause abrupt performance degradation of PMAC. The main reason why BTPS does not deliver all high priority packets is that the total bandwidth demands of high priority flows in some area of the network exceed the channel capacity.

Fig. 12 presents the aggregate throughput comparison between BTPS and PMAC for the generated random sce-

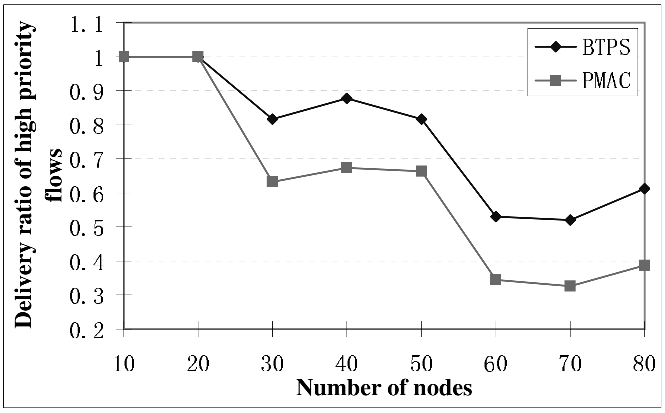

Fig. 11. Delivery ratio of high priority packets in random scenarios, comparison between BTPS and PMAC

narios. With an increase in the number of flows in the 1000m×1000m area, the contention degree for each flow tends to become higher. BTPS ensures high priority packets' delivery first, then low priority packets use as much bandwidth as possible after satisfying requirements of the high priority flows. On the other hand, PMAC lacks the capability to resolve contention among high priority flows efficiently under situations with high degree of contention; also the low priority packets cannot make full use of available bandwidth due to larger "interframe space" duration. For these reasons, it is not surprising that the proposed BTPS protocol provides higher aggregate throughput than PMAC [8]. In the case of 80 nodes, BTPS gains 58% aggregate throughput over PMAC.

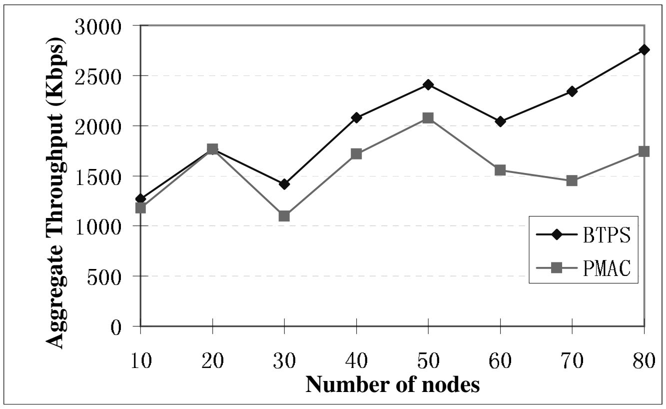

Fig. 12. Aggregate throughput comparison between BTPS and PMAC in random scenarios

VI. Conclusion

We present a priority scheduling MAC protocol (BTPS) for ad hoc networks. With the use of two narrow-band busy tone signals, BTPS ensures channel access of high priority packets. Furthermore, in the absence of high priority packets, low priority flows can make full use of available bandwidth in BTPS. Simulation results demonstrate the effectiveness of BTPS protocol with respect to "delivery ratio of high priority packets" and "aggregate throughput".

References

[1] ETSI TC-RES, "Radio Equipment and Systems(RES); HIgh PErformance Radio Local Area Network(HIPERLAN) Type 1; Functional specification," *European Telecommunication Standard ETS 300 652*, Oct 1996.

[2] IEEE, "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," *IEEE Standard 802.11*, June 1999.

[3] Andrew. Muir and J.J. Garcia-Luna-Aceves, "An Efficient Packet Sensing MAC Protocol for Wireless Networks," *ACM Journal on Mobile Networks and Applications*, vol. 3, no. 2, pp. 221–234, August 1998.

[4] 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, Autumn 1996.

[5] J.L.Sobrinho and A.S.Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, August 1999.

[6] Chunhung Richard Lin and Mario Gerla, "Real-time support in multihop wireless networks," *Wireless Networks*, vol. 5, pp. 125–135, 1999.

[7] Dr-Jiunn Deng and Ruay-Shiung Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method," *IEICE Trans. Commun.* , vol. E82-B, no. 1, January 1999.

[8] Imad Aad and Claude Castelluccia, "Differentiation mechanisms for IEEE 802.11," *IEEE INFOCOM*, April 2001.

[9] Michael Barry, Andrew T. Campbell, and Andras Veres, "Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks," *IEEE INFOCOM*, April 2001.

[10] V.Kanodia, C.Li, A.Sabharwal, B.Sadeghi, and E.Knightly, "Distributed Multi-Hop Scheduling and Medium Access with Delay and Throughput Constraints," *MOBICOM*, August 2001.

[11] J.Deng and Z.J.Haas, "Dual busy tone multiple access(DBTMA): A new medium access control for packet radio networks," *Proc. IEEE ICUPS'98*, Oct 1998.

[12] Fouad A. Tobagi and Leonard 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. COM-23, no. 12, December 1975.

[13] VINT Group, "UCB/LBNL/VINT network simulator ns (version 2)," *http://www.isi.edu/nsnam/ns*.

[14] The CMU Monarch Project, "Wireless and mobility extension to ns," *http://www.monarch.cs.cmu.edu*.