Lower-Priority-Triggered Distributed MAC-layer Priority Scheduling in Wireless Ad Hoc Networks

Wei Zhou and Turgay Korkmaz
Computer Science
University of Texas at San Antonio
San Antonio, Texas 78249
Email: korkmaz@cs.utsa.edu

***Abstract*— One of the key issues in supporting real-time and mission-critical applications on wireless ad hoc networks is how to honor the prioritization requirements of different flows at the MAC-layer while being fair to the flows within the same priority level. In response to this, researchers have proposed various MAC-layer prioritization mechanisms. These mechanisms try to allocate more bandwidths to the higher priority flows than that to the lower priority flows. However, this is often not enough to meet the requirements of higher priority flows, particularly when the low priority traffic is overwhelming. In this paper, we propose a fully distributed MAC-layer priority scheduling mechanism, called LPT-DPS (Lower-Priority-Triggered Distributed Priority Scheduling), that can allocate channel bandwidth to flows with different priorities in such a way that the higher the priority level of the flow, the higher the bandwidth that the flow acquires. Using extensive simulations, we demonstrate the effectiveness of the proposed LPT-DPS in delivering the high priority traffic even when the low priority traffic is overwhelming.**

I. Introduction

Supporting real-time audio/video applications and mission-critical applications on wireless ad hoc networks is an increasingly important and yet a challenging task, as it requires the underlying networking protocols to be empowered with appropriate Quality of Service (QoS) mechanisms. In response to this, the research community has been extensively investigating various QoS-related issues at the application, transport, network, data link (Medium Access Control (MAC)), and physical layers [1], [2], [3]. Accordingly, much work has been done on QoS, particularly in the context of IEEE 802.11 and its infrastructure mode [4]. Some work has also been done on cross layer issues for providing QoS support in wireless ad hoc networks. However, the existing solutions are not yet complete or being integrated. Moreover, they provide, at best, some level of QoS differentiations rather than QoS guarantees.

As an important and basic QoS mechanism, we focus on providing traffic prioritization in wireless ad hoc networks at the MAC layer. Traffic prioritization needs to be performed within and between wireless nodes. Accordingly, we divide this task into two parts: intra- and inter-prioritization. By intra-prioritization, we mean that a given node (say u) should be able to transmit higher-priority packets before lower-priority ones in itself. By inter-prioritization, we mean that node u should be able to transmit higher-priority packets before lower-priority ones in its neighbors. In essence, intra-prioritization can easily be achieved by using existing priority queuing and scheduling mechanisms in each node. However, due to shared nature of the underlying wireless channel and its distributed access control, it is not easy to guarantee that a high-priority packet at node u can be sent before a low-priority packet in the neighbors of node u. In the rest of this paper, we focus on how to provide inter-prioritization at the MAC layer and briefly refer to it as prioritization.

In the literature (see Section II and references therein), researchers proposed various approaches to deal with traffic prioritization issues in wireless networks. In general, these efforts focus on tuning some parameters of the underlying MAC-layer protocols such as contention window, back-off algorithm and interval, and inter-frame space etc. These techniques may provide some level of differentiation and better than best-effort service, but not the prioritization guarantees that are expected in various mission-critical applications. Finally most of these techniques are proposed in the context of IEEE 802.11 and its infrastructure mode, and may not be directly applied in multi-hop wireless networks.

To provide the prioritization at the MAC layer, we need to change the standard distributed coordination function (DCF) of IEEE 802.11 with a *priority*-based distributed coordination function (P-DCF) that can transmit higher priority packets before lower priority ones in neighboring nodes. To achieve this, we propose a Lower-Priority-Triggered Distributed Priority Scheduling (LPT-DPS) scheme that integrates the following modifications into the existing DCF:
(1) Make the waiting time between RTS–CTS and CTS–DATA exchanges proportional to the priority level of the ongoing transmission;
(2) Trigger the nodes that have higher priority frames to interrupt the ongoing transmissions of low priority frames during RTS-CTS exchange. We give the details of the proposed scheme and its operation in Section III.

The rest of this paper is organized as follows. Section II introduces the related work. Section III describes the proposed priority scheduling scheme. Section IV shows the simulation results. We conclude this paper and give future work in Section V.

II. RELATED WORK

In general, existing prioritization mechanisms for IEEE 802.11 MAC layer can be categorized as shown in Figure 1. Synchronized schemes share a common characteristic that is

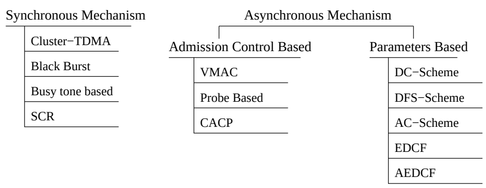


Fig. 1. Classification of prioritization mechanisms in wireless networks.

the access to the channel is usually done a TDMA fashion. For example, cluster-TDMA [5], the black burst scheduling (BTPS) scheme [6], and synchronous collision resolution (SCR) [7] can be put under this category. In contrast, asynchronous schemes share the channel in a random access manner. Asynchronous schemes can be further divided into two groups. The first group applies admission control policies on the traffic before it goes into the network. Under this category, we may list Virtual MAC (VMAC) [8], [9], [10] , Probe Based [11], andContention-aware Admission Control Protocol (CACP) [12]. The second group mainly manipulates various DCF parameters such as contention window, back-off algorithm and intervals, and inter-frame spaces to differentiate high priority traffic from low priority traffic. In the rest of this section, we mainly focus on these parameters-based schemes, as they constitute the most related work to our proposed scheme.

In general, existing parameters-based schemes adjust various parameters to increase the probability that a higher priority packets will be sent before the lower priority ones. For example, the DC mechanism in [13] makes modifications to the standard IFS waiting stage and backoff stage so that higher priority frames will have shorter IFS and backoff time than lower priority ones. Specifically, the IFS waiting time is equal to PIFS for stations (STAs) having high priority frames; and DIFS for STAs havign low priority frames. As for the backoff stage, the high priority STAs uses $\lfloor rand(0, 2^{2+i} - 1) \rfloor$ as their back time generation function, while the low priority ones use $\lfloor rand(2^{2+i}, 2^{3+i}) \rfloor$ as theirs. Since each one of the two stages can take one of two possible values, DC mechanism offers a total of 4 priority classes. Authors in [14] considered the similar ideas as in [13] with different time values for the IFS waiting stage and a different backoff algorithm.

The Distributed Fair Scheduling (DFS) scheme in [15] also manipulates the backoff interval (BI) to differentiate flows with different priorities. In DFS, The BI is computed as a function of packet size and the weight of the STA. The goal of the function is to give shorter BIs to higher priority packets so that they can be transmitted faster. DFS also borrowed the idea of self-clocked fair queuing in [16]. Accordingly, the fairness among flows is achieved by taking the packet size into the computation of BI, so that flows with smaller size packets could send more often. It is shown in [15] that DFS has higher throughput than standard IEEE 802.11 DCF. However, the main concern for DFS is its complexity in computing BI.

In [17] the authors summarized three techniques to differentiate traffic according to its priorities. The first one is to change the DCF backoff function so that the function increases with a different rate for flows with different priorities. The goal here is to make the contention window (CW) of higher priority flows increase slower. The second technique is to use different DIFS values for different priorities. For example, STAs with priority j would have a DIFS value of $DIFS_j$, and $DIFS_{j+1} < DIFSj$ means that STAs with priority $j+1$ tend to start transmission earlier than those with priority j. In addition, to avoid collisions among the same priority frames, they maintain the backoff mechanism in such a way that the maximum CW size added to $DIFS_j$ is $DIFS_{j-1} - DIFS_j$. The objective of their last mechanism is the maximum frame length, that is, STAs with higher priorities have larger preset maximum frame lengths, therefore, the higher the priority a STA has, the more information it transmits per medium access. However, this mechanism is susceptible to a noisy channel, because large frames are more likely to be corrupted than shorter ones under such an environment.

To provide traffic prioritization, the recently standardized IEEE 802.11e has adopted enhanced DCF (EDCF) [18] as one of the two access methods of 802.11e Hybrid Coordinate Function (HCF)[1]. EDCF uses a different IFS called arbitration inter-frame space (AIFS), and supports up to eight user priorities. The user priorities are mapped to four access categories (ACs), and each AC is associated with a queue that possesses a set of AIFS and CW values of its own, i.e., AIFS[AC], $CW_{min}[AC]$, and $CW_{max}[AC]$. According to the mapping between user priorities and access categories, the frame is put into one of the four queues. During the contention process, EDCF uses AIFS[AC], $CW_{min}[AC]$, and $CW_{max}[AC]$, instead of DIFS, CW_{min}, and CW_{max} of the DCF. If two access categories in the same STAs want to access the channel at the same time, then an internal scheduler of EDCF will resolve this virtual collision by always granting the access to the AC with higher priority. However, external collisions, meaning the collisions between STAs, may still occur.

The Adaptive EDCF (AEDCF) proposed in [19] tries to introduce dynamic factors into the basic EDCF mechanism so that the parameters of each AC queue can reflect the dynamicity of the channel conditions. In AEDCF, once a frame is successfully transmitted, the CW of that queue is not reset to $CW_{min}[AC]$ as usual. Instead, it is set to a value that is related to an estimated collision rate, f, in such a way that the larger the f, the greater the $CW_{min}[AC]$.

Problems with the parameters-based schemes: The main challenge for the asynchronous schemes is that they cannot guarantee the demanded bandwidth by the higher priority flows when the lower priority traffic is overwhelming. For instance,

[1]In 802.11e, EDCF is also known as HCF contention-based channel access (EDCA).

for the parameters based schemes, a lower priority flow may get a smaller backoff number (although with low probability) than its higher priority peer. Thus, it may transmit its low priority frame before a high priority one. With the increase of low priority flows, the probability of low priority frames being transmitted earlier than high priority ones becomes larger. This causes a problem, that is, when the higher priority flows are overwhelmed by the lower ones, the former cannot get enough bandwidth that they need. We show this in Section IV, using the DC mechanism in [13] as an example.

III. Priority-based Distributed Coordination Function (P-DCF)

Providing traffic prioritization in wireless networks is a challenging problem, particularly in the case of distributed random access protocols such as the Distributed Coordination Function (DCF) mode of IEEE 802.11. As we reviewed in Section II, existing solutions consider tuning various parameters such as contention window, back-off algorithm and interval, and inter-frame spaces that are used by the DCF mode of IEEE 802.11. These techniques may provide some level of differentiation and better than best-effort service. However, they cannot guarantee providing traffic prioritization as requested by mission-critical applications. In this Section, we specifically investigate how to achieve traffic prioritization in the case of the DCF mode of IEEE 802.11. For this, we modify the existing DCF of IEEE 802.11 and accordingly develop Priority-based Distributed Coordination Function (P-DCF) for IEEE 802.11.

In this section, we first give a brief introduction to IEEE 802.11 protocol and its DCF mode. Then describe our proposed scheme and illustrate its operations.

A. Brief Introduction to IEEE 802.11 protocols

IEEE 802.11 [20] series protocols are the *de facto* standards in wireless networks. IEEE 802.11 series protocols define two medium access modes, namely an optional point coordination function (PCF) and a mandatory contention-based distributed coordination function (DCF). We mainly discuss DCF mode, where the channel access is based on carrier sensing multiple access with collision avoidance (CSMA/CA).

DCF mode has two carrier-sense mechanisms, namely physical and virtual carrier-sense. The physical carrier-sense function is provided by the physical layer, i.e., the STA's transmitter. The virtual carrier-sense function is usually referred to as the network allocation vector (NAV). Each STA maintains a NAV counter whose value is set by copying the value of the duration information field of each overheard frame. This counter is counted down at a uniform rate. When it reaches zero, the virtual carrier-sense indicates that the medium is idle; otherwise, it indicates that the medium is busy.

Typically, in DCF access mode, before transmitting a frame, a STA must sense an idle channel for a period of time to make sure no other STAs are transmitting. This period of time can be generally divided into two stages. The first stage is called interframe space (IFS) waiting stage, the length of which is the value of DCF interframe space (DIFS). The second stage is known as the backoff stage, whose length is determined by a backoff algorithm using the current contention window (CW) value as its input (We will describe the backoff algorithm and CW shortly). If at any time during the waiting period, the STA senses that the channel is busy, it must defer its intended transmission until the end of the on-going transmission. If the channel is still idle at the end of the waiting period, the STA can start its transmission. At the receiver side, after successfully receiving a frame, the receiving node waits for a period of short interframe space (SIFS) time, and then replies with an ACK frame.

Time on the channel is slotted in IEEE 802.11 series protocols. Each STA can send only at the start of some slot. The slot size is determined by the characteristics of the physical layer. In the backoff stage, the node selects a random backoff interval and decrements the backoff interval counter *only* when the channel is idle. Because the time is slotted in IEEE 802.11 protocols, the backoff interval is actually reflected by a random number between zero and the node's current CW value. When the backoff timer counts down, it counts down in terms of slot time. The value of CW reflects the current severity of contention for the channel, and CW value is increased by a binary exponential function. That is, each time an STA detects an unsuccessful transmission, it doubles its CW value until the value reaches a preset CW_{max}. After that, CW remains to be CW_{max} until either the data frame is successfully transmitted or the frame is dropped due to preset retransmission attempts has been exceeded. Then, the CW is reset to CW_{min} for the next to-be-transmitted frame.

IEEE 802.11 protocols define four IFSs. Besides the above mentioned DIFS and SIFS, the other two are PCF interframe space (PIFS) and extended interframe space (EIFS). PIFS is exclusively used in PCF mode; while EIFS is used in DCF mode whenever the MAC layer has detected a corrupted frame. The lengths of these IFSs are different, and the list in the ascending order of length is SIFS, PIFS, DIFS and EIFS. The different length values are designed to provide different priority levels for access to the channel. For example, the SIFS time is the shortest among other three IFSs, and is meant to give the highest precedence to ACK frames. Finally, to alleviate the potential negative affects by hidden node problems [21], the communication parties can optionally use the request-to-send/clear-to-send (RTS/CTS) mechanism that is also defined in DCF.

B. Proposed P-DCF Scheme

In general, the proposed P-DCF scheme is a modification of the existing DCF. First modification is simply to include the priority information into both control (e.g. RTS/CTS/ACK) and data frames. For instance, adding four extra bits to each frame can provide sixteen levels of priorities. This way, we make sure that both the destination station (STA) and other neighbor STAs that overhear the frames know what the priority level is for the ongoing transmission session. Second and more important modification is to employ a distributed priority scheduling (DPS) scheme so that higher priority packets can be sent before lower priority ones. For this purpose, we propose a Lower Priority Triggered DPS (LPT-DPS) mechanism that integrates two key mechanisms into the existing DCF:

- Make the waiting time between RTS–CTS and CTS–DATA exchanges proportional to the priority level of the ongoing transmission.
- Trigger the STAs that have higher priority frames to interrupt the ongoing transmissions of low priority frames during RTS–CTS exchanges.

The first mechanism can easily be realized as illustrated in Figure 2. Assume that the source STA *S* creates or receives a frame with priority *p* and wants to send it to the destination STA *D*.[2] As in the standard DCF, *S* first sends out RTS.

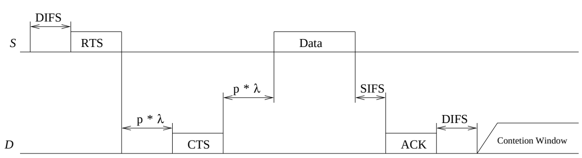


Fig. 2. Successful transmission of a packet with priority level *p*.

However, if *D* successfully receives the RTS, it waits for $p \times \lambda$ amount of time and then reply with CTS, where λ is a constant time that will be discussed later. Upon successful reception of CTS, *S* waits for $p \times \lambda$ amount of time and then send its DATA. Finally, after successfully receiving DATA, *D* waits for *SIFS* amount of time and then sends ACK back, as in the standard DCF.

The second mechanism along with the first one is essential to coordinating STAs in such a way that the STAs having the highest priority level frames will transmit first. Unfortunately, this cannot be achieved by using the existing DCF of IEEE 802.11 or other mechanisms in the literature, in which the frames of higher priority must wait until the current transmission session between *S* and *D* ends. In contrast to DCF, the proposed LPT-DPS scheme (i) interrupts the transmission of lower priority frames during RTS–CTS exchanges, and (ii) triggers the STAs that have the highest priority frames to start transmitting. To achieve these two tasks, every STA employs the following procedure:

Upon Receiving RTS Frame at the Destination STA

1) If the destination STA does not have a backlogged frame or if it has a backlogged frame with priority level lower than or equal to the received frame, the destination STA responds to the received frame as standard IEEE 802.11 protocol does. That is, the STA replies to the sender with CTS frame after waiting $p \times \lambda$ amount of time.
2) If the destination STA has a backlogged frame with priority level higher than the received frame, the STA drops this newly received RTS frame, and then waits for $p_{self} \times \lambda$ amount of time and then sends out its own frame that has higher level priority, where p_{self} is the priority level of its own backlogged frame. This way the transmission of a lower priority frame will be interrupted and another frame with higher priority level will be sent.

Upon Receiving CTS Frame at the Source STA

1) If the source STA receives the CTS frame from the destination, it first checks whether or not it is waiting for any ongoing higher priority transmission. If yes, the source STA simply ignores this CTS, otherwise, it waits for $p \times \lambda$ amount of time before starting data transmission. During this waiting period, if the source overhears any higher priority frame, it should backoff according to the duration information in that higher priority frame.

Upon Overhearing RTS or CTS at the Third STA

1) If the overhearing STA does not have a backlogged frame or if it has a backlogged frame with priority level lower than or equal to that of the overheard frame, the STA sets its NAV timer according to the duration information in the received frame, so that it will not interfere the ongoing transmission.
2) If the overhearing STA has a backlogged frame with priority level higher than the overheard frame, the STA does not set its NAV timer. On the contrary, it waits for $p_{self} \times \lambda$ time and then sends its frame that has higher level priority, where p_{self} is the priority level of its own backlogged frame. This way the transmission of a lower priority frame will be interrupted and another frame with higher priority level will be sent.

Note that when a station receives or overhears a DATA or ACK frame, the LPT-DPS will not interrupt the ongoing data transmission.

Collisions triggered by low priority level frames

In the above discussion, an RTS or CTS frame with low priority level may trigger the transmission of high priority frames. Since the waiting time for a high priority STA to start its transmission is $p_{self} \times \lambda$, multiple neighbors with the same high priority level start transmitting simultaneously after waiting for $p_{self} \times \lambda$ time after overhearing a low priority RTS or CTS, causing collisions. To cope with this problem, we introduce randomness to the starting time of transmissions of higher priority level frames. Specifically, after the above mentioned $p_{self} \times \lambda$ waiting time, the STAs with high priority frames wait for an extra random amount of time. For this, we divide the λ time into m time slots, as shown in Figure 3. Each

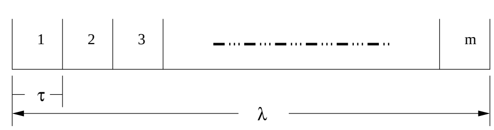


Fig. 3. Dividing λ into m slots of τ time.

slot is of τ time, where τ is the maximum propagation delay

[2]Note that the lower the value of p, the higher the priority level in our scheme.

in the system. Each STA with high priority frame selects a random number x between $[0, m)$ and then waits for $x \cdot \tau$ time. This way, if one of the STAs starts transmission at time t_0, then its neighbors will sense the signal by the time $t_0 + \tau$, and thus they delay their own transmissions to the next session. In case of two STAs' selecting the same random numbers, there would be a collision. This maybe resolved in the next iteration of the protocol. However, this collision rate might be excessive if the number of triggered STAs (say n) with the same priority level is large. To avoid high collision rate, each eligible STA needs to also take the value of n into account when deciding in which time slot to start transmission.

Assume that the triggered STAs have a way of determining the value of n (e.g., in the worst case, n will be equal to the number of neighbors of a STA and that approach is used in our simulations; but we can also try to better estimate it by maintaining an additional priority re-transmission counter and increasing it as we see collisions during this extra waiting time). Now we make each eligible STA listen to the channel and (if the channel is idle) start transmission at the beginning of a time slot with probability q. Our goal is to determine the value of q such that the successful transmission rate can be maximized. Given that we have m time slots as mentioned above, n eligible STAs of each independently starts transmitting at the beginning of a time slot with probability q, we can easily determine the probability for a single STA to successfully transmit in the i^{th} time slot as follows:

$$((1-q)^n)^{i-1} q(1-q)^{n-1}.$$

Since we have m time slots, the probability for a single STA to successfully transmit will be

$$q(1-q)^{n-1} \sum_{i=0}^{m-1} ((1-q)^n)^i$$

which is equal to

$$q(1-q)^{n-1} \frac{1-(1-q)^{nm}}{1-(1-q)^n}.$$

Considering n stations, the overall success probability will be

$$S(q) = nq(1-q)^{n-1} \frac{1-(1-q)^{nm}}{1-(1-q)^n}.$$

To determine the value of q that maximizes $S(q)$, we need to solve

$$S'(q) = \frac{vu' - uv'}{v^2} = 0,$$

where

$$u = nq(1-q)^{n-1}(1-(1-q)^{nm}),$$

$$v = 1-(1-q)^n,$$

$$u' = n(1-q)^{n-1} - nq(n-1)(1-q)^{n-2} - n(1-q)^{nm+n-1} + nq(nm+n-1)(1-q)^{nm+n-2},$$

$$v' = n(1-q)^{n-1}.$$

Unfortunately, it is not an easy task to analytically determine the roots of such a function. Instead, we use Newton's method [22]. To solve $S'(q) = 0$, Newton's method has the following formula

$$q_{j+1} = q_j - \frac{S'(q_j)}{S''(q_j)} \quad j = 0, 1, 2, \ldots,$$

where q_0 is given as an initial guess (in our case we set $q_0 = 0.01$). As seen in the above formula, Newton's method requires us to also take the second derivative of $S(q)$. This can easily be done as follows.

$$S''(q) = \frac{v^2(vu'' - uv'') - (vu' - uv')2vv'}{v^4},$$

where

$$u'' = -n(n-1)(1-q)^{n-2} - n(n-1)(1-q)^{n-2} + nq(n-1)(n-2)(1-q)^{n-3} + n(nm+n-1)(1-q)^{nm+n-2} + n(nm+n-1)(1-q)^{nm+n-2} - nq(nm+n-1)(nm+n-2)(1-q)^{nm+n-3},$$

$$v'' = -n(n-1)(1-q)^{n-2}.$$

In our case, Newton's method is easier than solving the original function and it converges quickly. Also we need to just compute optimal value of q once for different n and m values and incorporate these values into the protocol, as constants. For example, when $m = 5$, the values of q should be 1 for $n = 1$, 0.2529 for $n = 2$, 0.1630 for $n = 3$, 0.1205 for $n = 4$, 0.0957 for $n = 5$, 0.0794 for $n = 6$, 0.0678 for $n = 7$, 0.0592 for $n = 8$, 0.0525 for $n = 9$, 0.0472 for $n = 10$, and so on. Figure 4 illustrates the optimal values of q and the corresponding overall success probability (i.e., $S(q)$).

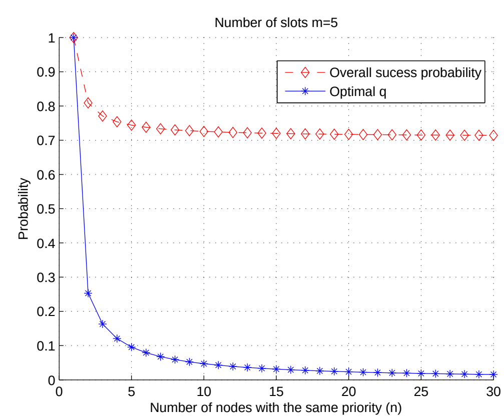


Fig. 4. Optimal values of p and corresponding success probability.

It is also worth noting that the overall success depends not only on the value of λ, τ, and the number of triggered neighbor STAs, but also on the geographical distribution of the STAs having high priority frames. For example, in Figure 5, suppose S_1 and S_2 have frames with the same priority levels, and both of them are triggered by the low priority transmission of S_0. In

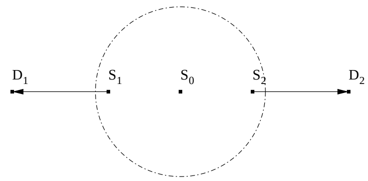


Fig. 5. No collisions at D_1 or D_2

this case, because S_1 and S_2 are located too far from D_2 and D_1 respectively to interfere the two receivers' signal reception, hopefully no collision would be detected at either receivers.

C. Operations of the Protocol

In this section, we give a simple example to show how LPT-DPS operates. Consider Figure 6. Station A wants to send

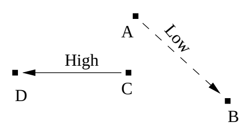


Fig. 6. A wireless ad hoc network with high and low priority flows.

a low priority frame to station B, and station C has a high priority frame backlogged for station D. To illustrate how our protocol behaves differently from others, we assume that A starts its RTS first. Under other MAC protocols including IEEE 802.11, A's RTS seizes the channel and allows it to transmit its frame to B without being interrupted. Meanwhile C must wait until the transmission between A and B is finished, even if it has a frame with higher priority to transmit. The frame exchange sequence in this case is illustrated in Figure 7.

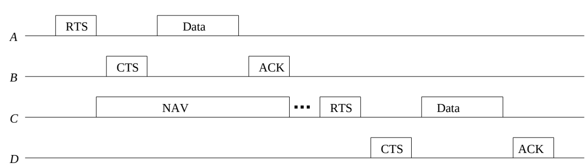


Fig. 7. The operation of existing MAC protocols.

Figure 8 illustrates the sequence in our protocol. Once C overhears the RTS with lower priority, it waits for $p_{self} \times \lambda$ amount of time, and then starts to transmit its own RTS. When

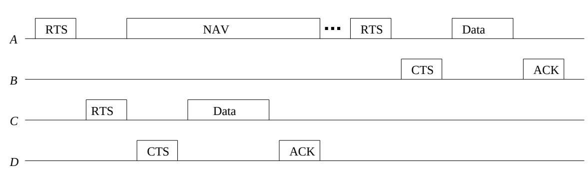


Fig. 8. The operation of the proposed protocol.

A overhears this RTS, it delays its own transmission according to the duration information in C's RTS, because it realizes that C's RTS has higher priority than its own. If B can overhear C's RTS too, B drops its CTS. If B cannot overhear C's RTS, B sends to A a CTS frame, which will be dropped by A. Either way, the transmission between C and D will proceed first. After that, A restarts its transmission with B.

D. Sensing Range Problem in Providing Prioritization

When all the STAs are in the communication range of their neighbors, the proposed scheme guarantees to transmit higher priority frames before the lower priority ones. However, in practice, some STAs might be in the sensing range of some other STAs. In this case, a STA within the sensing range of some transmitting neighbors will not be able to decode the signal coming from such neighbors. In other words, a STA will not understand what is the priority level of the ongoing transmission. To deal with the nodes in sensing range, we currently use the same strategy in the standard IEEE 802.11 protocol, where once a STA senses but cannot decode the signal, it should first wait until the channel is idle. It then waits for an extra EIFS amount of time before starting its own transmission. This guarantees that sensing STAs will not interrupt the on-going transmissions. In P-DCF or other existing mechanisms, this situation may cause low priority traffic to delay the transmission of some high priority traffic. To illustrate this, let us consider the example in Figure 9. Assume that station *A* wants to send a low priority frame to

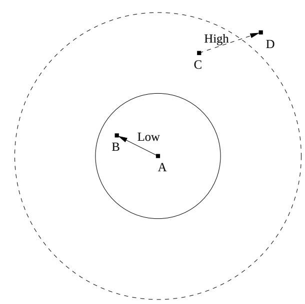


Fig. 9. The sensing range problem.

B while station *C*, which is located within the sensing range of *A*, wants to send a high priority frame to *D*. If *A* starts transmitting RTS earlier than *C*, because *C* can only sense but not decode the signal, *C* cannot know the priority level of *A*. So *C* has to wait. In this case, the high priority traffic between *C* and *D* is delayed by the low priority traffic between *A* and *B*. Using simulations in the next section, we show that the proposed scheme provides much better performance than existing schemes under sensing range problem. The effect of sensing range problem will be minimized as the technology improves and reduce the sensing range while increasing the communication range of the underlying hardware.

IV. Performance Evaluation

We implemented our LPT-DPS scheme by making modifications to ns-2 network simulator [23]. In addition, we implemented the DC mechanism in [13], as a representative of parameters-based schemes. We then compare LPT-DPS and DC mechanism with the standard IEEE 802.11 protocols [20].

We use the default radio interface model in ns-2, which approximates the first generation WaveLan radio interface with 1 Mbps bit rate[3] and 250 meter transmission range using omnidirectional antenna. We also integrate the NO Ad-Hoc (NOAH) routing agent [24] into ns-2 to provide static routing in our experiments whereby we eliminate the bandwidth overheads caused by routing algorithms. We use constant bit rate (CBR) traffic flows with packet size of 512 bytes in all the experiments. In our simulations, the λ is equal to the SIFS value in ns-2, which is 10 μs. The maximum propagation delay, τ, is equal to 2 μs.

We conduct experiments on both single-hop and multi-hop topologies. In case of single-hop topologies, all nodes are randomly distributed in a circle area such that all the nodes are within the communication range of each other (so there is no sensing range problem). In this case, our scheme guarantees that higher priority flows will access the channel before the lower priority flows and get the desired level of prioritization. In case of multi-hop topologies, nodes are randomly distributed in an 1000m x 1000m area and some nodes may not hear some other nodes or fall in sensing range of some other nodes. In this case, due to sensing range problem, the proposed scheme may not guarantee that the higher priority flows access the channel before lower priority ones. But simulation results show that our scheme allocates almost all the channel capacity to higher priority flows.

A. Single-hop Topologies

To clearly demonstrate the effectiveness of our scheme, we first consider three pairs of STAs that are randomly distributed in a circle area. We set up three flows among these three pairs of STAs, each with different priority of 2, 3, and 4, respectively. We increase the flow rates of the three flows from 20 kbps up to 700 kbps. The simulation results we show here are averages over 32 runs, and each simulation run lasts for 120 second duration. The performance metrics we use include both total flow throughput and average packet delay.

The throughput of the compared three schemes are shown in the first column of Figure 10. In Figure 10(a1), we show the performance of the standard IEEE 802.11 protocol, where no priority mechanism is considered. In this case, the three flows start to share channel bandwidth with almost the same ratio, as expected. In Figure 10(b1), we show the performance of our proposed scheme. In this case, the channel bandwidth is always used to first satisfy the demand of the highest priority flow, and then the remaining bandwidth is used to satisfy the demand of the second highest priority flow and so on. As the channel is saturated, the throughput of the three flows goes down one by one in the descending order of their priority levels. As a result, the lowest priority flow gets only the bandwidth remaining from higher priority flows. In Figure 10(c1) we show the performance of DC mechanism. It differentiates flows based on their priorities and gives

[3] Because of the MAC layer overheads, such as RTS/CTS/ACK frames, MAC frame header infomation, and various IFSs etc., the actual bandwidth seen by the application layer is less than 1 Mbps.

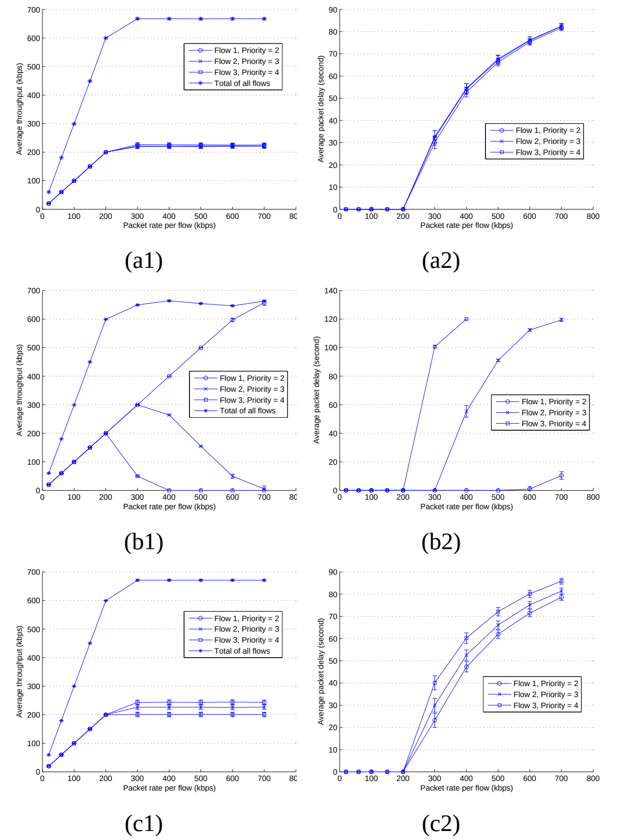
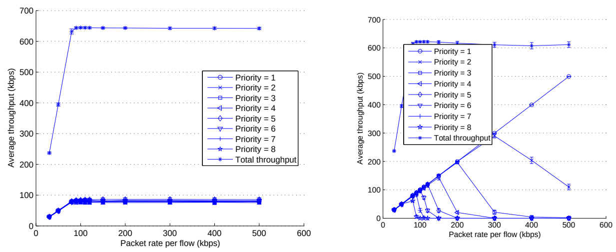


Fig. 10. Throughput (first column) and average delay (second column) in IEEE 802.11, LPT-DPS, and DC Mechanism, respectively.

more bandwidth to higher priority flows. However, when the network is overloaded (i.e. when the flow rate is above 200 kbps), the highest priority flow has only about 17% more bandwidth than the lowest priority flow. Although this is better than the standard IEEE 802.11 protocol in terms of delivering high priority frames, it does not meet the demand of the highest priority flow as much as in LPT-DPS. One thing worth noting is that the line of total throughput in Figure 10(b1) goes down a little, i.e., a small amount of bandwidth is wasted at the cost of providing the desired levels of prioritization to different flows.

The average packet delay for the compared three different mechanisms are shown in the second column of Figure 10. In Figure 10(a2), we show the delay for standard IEEE 802.11 protocol, where no priority mechanism is considered. For all flows, the delay is negligible before the channel is saturated. However, it begins to increase for all flows once the channel cannot hold all the traffic. Figure 10(b2) shows that LPT-DPS scheme allows the highest priority flow to capture the channel and encounter a very small delay until its own rate is above the channel capability. The second highest priority flow starts to experience long packet delays when the flow rate is at 400 kbps, because at that point the sum of the priority 2 flow and the priority 3 flow exceeds the channel capacity. As for the lowest priority flow, it cannot send a single packet during channel saturation because all the channel bandwidth is used up by the high priority flows. Figure 10(c2) shows the average delay metric for the DC mechanism. When traffic rate per flow is equal to or more than 300 kbps, again we see some differentiation between flows. However, the highest priority flow does not receive the desired service as much as in LPT-DPS.

1) Multiple Priority Levels: In Figure 11 we further evaluate the performance of LPT-DPS using 8 priority levels. Since



(a) (b)

Fig. 11. Throughput using eight priority levels in 802.11 and LPT-DPS, respectively.

the DC mechanism does not support this many priority levels, we simply compare the throughput of LPT-DPS with that of IEEE 802.11. In this case, we randomly distribute eight pairs of STAs in a circle such that all the sixteen STAs are within communication range of each other. Then each pair of STAs carries one flow with priority level from 1 to 8, respectively. We increase the flow rates of the eight flows from 30 kbps up to 500 kbps. From Figure 11(a), we see that all the flows almost equally share the channel bandwidth under the standard IEEE 802.11 protocol. However, as seen in Figure 11(b), LPT-DPS first gives the channel bandwidth to the flow with highest priority level, then to the flow with the second highest priority level, and so on. Even under 8 priority levels, LPT-DPS is still able to provide the desired level of priorities to different flows.

2) Multiple High Priority Flows with One Overwhelming Low Priority Flow : In this scenario, we consider the performance of the three schemes under multiple flows with the same priority levels and high volume of low priority traffic. We randomly place eleven pairs of STAs in a circle. Among the eleven pairs STAs, one pair forms a low priority flow with priority 4, and this flow has an overwhelming rate of 800 kbps in order to saturate the channel. Meanwhile, we have five flows of priority 2 and five flows of priority 3, respectively. We increase the flow rate for the two higher priority flows gradually from 15 kbps up to 150 kbps.

We measure the flow throughput for the lowest priority flow as well as the total flow throughput for the other two high priority flows. Figure 12(a) shows the results for IEEE 802.11 protocol. When the rate of the two high priority flows adds up, the ratio of the bandwidth that the lowest priority flow occupies decreases. Eventually, since standard IEEE 802.11 does not provide any prioritization, all the eleven flows share almost the same ratio of the channel capacity. Note that the lines for the two high priority flows in Figure 12 represent the total throughput of five flows.

Figure 12(b) presents the result for LPT-DPS mechanism, we see that the background traffic quickly drops to zero because higher priority flows demand and use more bandwidth than the lowest priority flow. In LPT-DPS, high volume low priority flow does not intend to compete channel bandwidth with the high priority flows, instead, because the triggering mechanism, it serves as a helper to transmit the high priority traffic as much as possible.

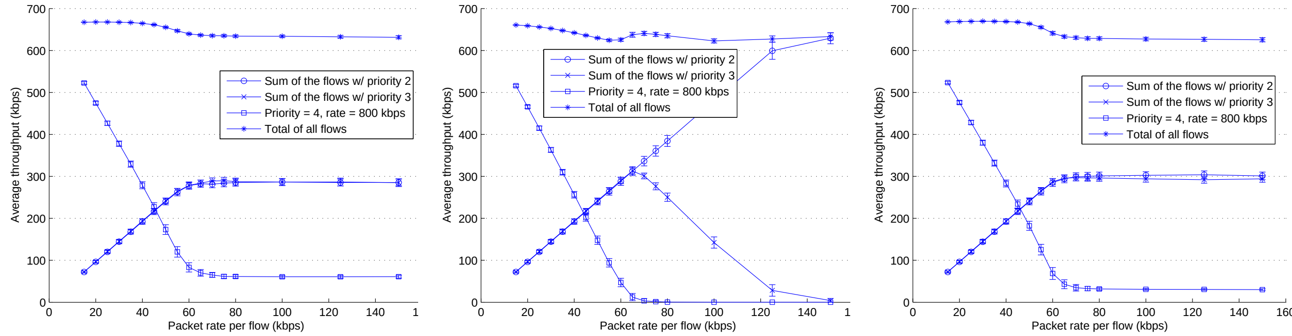
As shown in Figure 12(c), the DC mechanism tends to allocate more bandwidth to the higher priority flows because higher priority flows have larger probabilities to transmit their traffic. As a result, when the two high priority flows reach their maximum transmitting capability, the lowest priority flow has 50% less share of bandwidth than that in standard IEEE 802.11, which means the higher priority flows claim more bandwidth. However, the highest priority flow in DC mechanism does not get all the bandwidth it needs.

B. Multi-hop Topologies

In this experiment, we again compare three schemes (the proposed LPT-DPS, the DC mechanism and the standard IEEE 802.11 protocols) in terms of packet delivery effectiveness under multi-hop topologies, where sensing range problem appears. In a multi-hop topology, nodes are randomly deployed in an 1000m x 1000m area and some nodes may not hear each other or may fall into sensing range of others. As we focus on MAC layer, we first consider single-hop flows, the source and destination of each flow are selected from neighbor nodes. We later consider multi-hop flows, where the source and destination of each flow may not be direct neighbors.

Multi-hop Topologies, Single-hop Flows: We measure 6 cases, where the total number of flows in the 1000m x 1000m area is 4, 8, 16, 32, 64, and 128. In each case, half of the flows are high priority flows, and the other half are low priority flows. We use two priority levels: level 2 for the high priority flows and level 3 for the low priority flows. In our experiment, both the high and low priority flows bear the same traffic rate of 60 Kbps. For each of the 6 cases, we run our programs 20 times, and each simulation run is for 60-second duration.

Figure 13 shows the total throughput of all the flows and the total throughput of all the high priority flows together in the same figure. Thus, it is easy to see that, for each of the three schemes in the figure, the gap between the total throughput and the corresponding high priority throughput is the total throughput of all the low priority flows for that scheme. For all the three schemes, the total throughput decreases along with the traffic increases. This is mainly because more and more collision happens in the multi-hop wireless network. The total throughput of LPT-DPS is slightly less than the other two schemes. The main reason for LPT-DPS having lower total throughput is due to the fact that a low priority RTS or CTS in LPT-DPS is either dropped or ignored after triggering a high priority transmission, causing bandwidth loss. On the other hand, LPT-DPS performs much better than the other two



(a) (b) (c)

Fig. 12. Throughput in IEEE 802.11, LPT-DPS, and DC Mechanism, respectively.

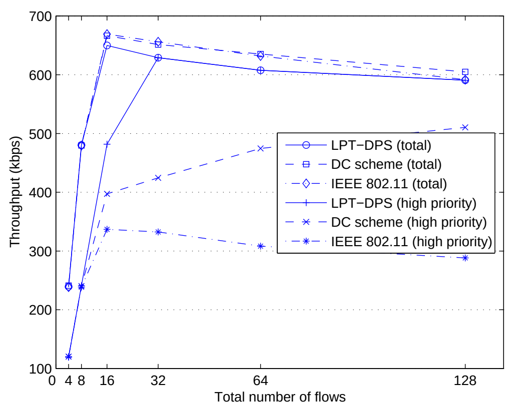


Fig. 13. Throughput comparison under multi-hop topologies and single-hop flows.

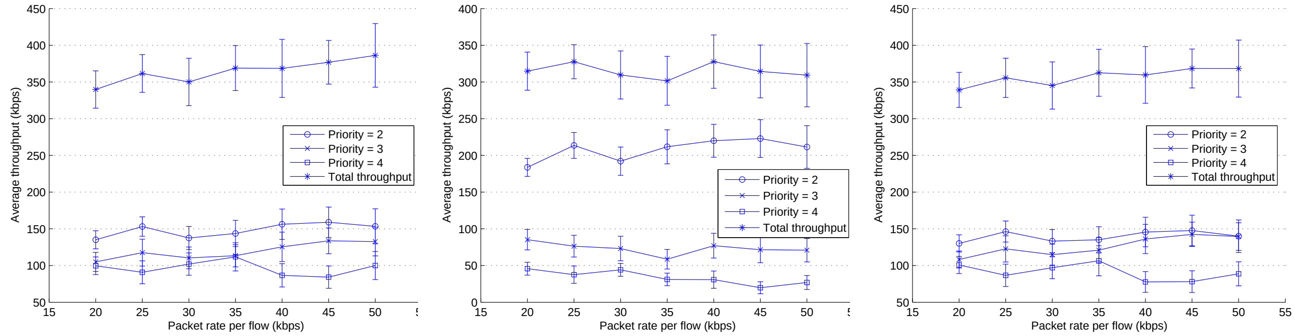
schemes in terms of delivering high priority traffic. Take the case of 16 flows for example, in LPT-DPS, the total throughput for all the 8 high priority flows is 481.87 kbps, which means each high priority flow gets the bandwidth it demands despite of the contending low priority traffic. Meanwhile, DC scheme delivers about 83% of the high priority traffic, and IEEE 802.11 can only delivers about 70%. When the traffic load keeps increasing, LPT-DPS allocates almost all the bandwidth to the high priority flows.

Multi-hop Topologies, Multi-hop Flows: We first deploy 100 nodes randomly in the 1000m x 1000m area. We then set up 10 random flows for each priority level 2, 3 and 4. Thus, there is a total of 30 flows in the covered square area. Note that the source and destination of each flow are randomly selected from the 100 nodes, and static routes are established between the sources and the corresponding destinations. We gradually increase the flow rate from 20 Kbps to 50 Kbps with a pace of 5 Kbps. We run our programs 32 times, and each simulation run is for 60-second duration. Note that we also made modifications to the queuing algorithms in all the 3 schemes so that they always forward the higher priority packets before lower priority packets, and when the queue is full, the lower priority packets get dropped first. This is to make sure that intra-prioritization is provided in all three schemes.

Figure 14(a), (b), and (c) show our simulation results for the standard IEEE 802.11, the LPT-DPS, and the DC. Compared to other 2 schemes, the LPT-DPS can better differentiate the flows with different priorities in terms of throughput by always giving more than 50% of total bandwidth to the highest priority traffic. Moreover, when the rate of low priority flows increase, the bandwidth used by the highest priority also increases because low priority flows trigger more transmissions of the highest priority packets, until after some point where the triggering causes the highest flows to compete for the channel among themselves. On the other hand, the DC mechanism does a little traffic differentiation while the IEEE 802.11 standard does almost none. In Figure 14(a), the priority level 2 traffic gets the most bandwidth even though IEEE 802.11 does not do any priority scheduling. But the reason is more likely because we always forward the highest packets in the network, and because the network is overloaded, thus, the queue at each node is almost always full so that low priority packets (i.e. priority 3 and 4) are mostly dropped in the middle of their way to their destinations. Another fact we note is that in DC and standard 802.11 cases the total throughput of the network is higher than that of the LPT-DPS. As we mentioned before, this is because the bandwidth used for the low priority RTS/CTS is wasted when this low priority RTS/CTS triggers another high priority frames in LPT-DPS.

V. Conclusions and Future Work

This work has proposed a fully distributed priority scheduling mechanism, LPT-DPS, for wireless MAC layer. LPT-DPS differs from other asynchronous schemes in how they treats low priority RTS/CTS frames. In LPT-DPS, RTS/CTS is used not only to reserve the floor to preclude neighbor traffic with the same priority, but also to trigger the transmission of higher priority traffic. Unlike in other schemes, where the transmission of high priority traffic can be delayed by low priority RTS/CTS, LPT-DPS makes sure that higher priority traffic interrupts the RTS/CTS exchange of low priority traffic. We have shown in the simulations that LPT-DPS delivers more high priority packets than the DC scheme [13] and standard



(a) (b) (c)

Fig. 14. Throughput in IEEE 802.11, LPT-DPS, and DC Mechanism under multi-hop topologies, multi-hop flows, respectively.

IEEE 802.11. Another advantage of LPT-DPS is that it can be easily combined with other QoS aware schemes, because LPT-DPS is a reactive mechanism and takes effect when a RTS/CTS has been sent out to the channel.

Our future work mainly includes the following two aspects. First, we plan to research the fairness issue among multiple flows with the same priority. As we described in Section III-B, if a flow is surrounded by multiple other flows with the same high priority, this low priority flow may trigger multiple high priority flows at the same time, thus, it is desirable to make sure each high priority flow has approximately the same share of the bandwidth. Second, we plan to explore the performance issues under mobility while also considering new QoS metrics such as jitters.

References

[1] D. D. Perkins and H. D. Hughes, “A survey on quality-of-service support for mobile ad hoc networks,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 503–513, Sep. 2002.

[2] P. Mohapatra, J. Li, and C. Gui, “Qos in mobile ad hoc networks,” *IEEE Wireless Communications*, vol. 10, no. 3, pp. 44 – 52, June 2003.

[3] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall, 2004.

[4] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, “A survey of quality of service in ieee 802.11 networks,” *IEEE Wireless Communications*, vol. 11, no. 4, pp. 6 – 14, August 2004.

[5] M. Gerla and T. C. Tsai, “Multicluster, mobile, multimedia radio network,” *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[6] X. Yang and N. H. Vaidya, “Priority scheduling in wireless ad hoc networks,” in *MobiHoc’02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2002, pp. 71–79.

[7] J. A. Stine and G. D. Veciana, “A paradigm for quality-of-service in wireless ad hoc networks using synchronous signaling and node states,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1301–1321, Sep. 2004.

[8] M. Barry, A. T. Campbell, and A. Veres, “Distributed control algorithms for service differentiation in wireless packet networks,” in *Proceedings of IEEE INFOCOM 2001*, vol. 1, Apr. 2001, pp. 582–590.

[9] A. Veres, A. T. Campbell, M. Barry, and L. H. Sun, “Supporting service differentiation in wireless packet networks using distributed control,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2081–2093, Oct. 2001.

[10] G. S. Ahn, A. T. Campbell, A. Veres, and L. H. Sun, “SWAN: service differentiation in stateless wireless ad hoc networks,” in *Proceedings of IEEE INFOCOM 2002*, vol. 2, Jun. 2002, pp. 457–466.

[11] S. H. Shah, K. Chen, and K. Nahrstedt, “Dynamic bandwidth management for single-hop ad hoc wireless networks,” in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003*, Mar. 2003, pp. 195–203.

[12] Y. Yang and R. Kravets, “Contention-aware admission control for ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 363–377, Jul.-Aug. 2005.

[13] D. J. Deng and R. S. Chang, “A priority scheme for IEEE 802. 11 DCF access method,” *IEICE Trans. on Commun.*, vol. E82-B, no. 1, January 1999.

[14] X. Pallot and L. E. Miller, “Implementing message priority policies over an 802.11 based mobile ad hoc network,” in *IEEE Military Communications Conference, 2001*, vol. 2, Oct. 2001, pp. 860–864.

[15] N. H. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless lan,” in *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom)*. New York, NY, USA: ACM Press, 2000, pp. 167–178.

[16] S. J. Golestani, “A self-clocked fair queueing scheme for broadband applications,” in *Proceedings of IEEE INFOCOM 1994*, vol. 2, Jun. 1994, pp. 636–646.

[17] I. Aad and C. Castelluccia, “Differentiation mechanisms for ieee 802.11,” in *Proceedings of IEEE INFOCOM 2001*, vol. 1, Apr. 2001, pp. 209–218.

[18] M. Benveniste, G. Chesson, M. Hoeben, A. Singla, H. Teunissen, and M. Wentink, “EDCF proposed draft text, IEEE working document 802.11-01/131r1,” Mar. 2001.

[19] L. Romdhani, Q. Ni, and T. Turletti, “Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks,” in *IEEE Wireless Communications and Networking*, vol. 2, Mar. 2003, pp. 1373–1378.

[20] “IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999), IEEE Standards for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.”

[21] 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, no. 12, pp. 1417–1433, December 1975.

[22] I. Bronshtein, K. Semendyayev, G. Musiol, and H. Muehlig, *Handbook of Mathematics*, 4th ed. Springer, 2004.

[23] “Network simulator - ns2,” www.isi.edu/nsnam/ns.

[24] “NO Ad-hoc Routing Agent,” http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/.