DISTRIBUTED MECHANISMS FOR QUALITY OF SERVICE IN WIRELESS LANs

WASAN PATTARA-ATIKOM AND PRASHANT KRISHNAMURTHY, UNIVERSITY OF PITTSBURGH
SUJATA BANERJEE, HEWLETT-PACKARD LABORATORIES

ABSTRACT

Wireless local area networks are gaining popularity at an unprecedented rate, at home, at work, and in public hot spot locations. As these networks become ubiquitous and an integral part of the infrastructure, they will increasingly be used for multimedia applications. There is limited QoS support in WLANs, which will become an impediment in deploying multimedia applications. In this article we present a tutorial on QoS support in IEEE 802.11 WLANs with a focus on the distributed MAC protocol of 802.11. Most QoS support mechanisms proposed for 802.11 use well-known techniques such as priority assignment and fair scheduling, and map QoS metrics into some existing 802.11 MAC parameter, thereby avoiding a redesign of the MAC protocol. We provide a taxonomy of the mechanisms and describe the essential concepts, problems, and advantages of each mechanism. From our study, we conclude that choosing the right set of MAC parameters and the QoS mechanism itself to provide predictable QoS in 802.11 networks is still an open problem.

INTRODUCTION

Wireless local area networks (WLANs) have increasingly become the edge network of choice. Concurrent with the expansion of WLANs is a high demand for quality of service (QoS)-sensitive (i.e., delay-constrained or throughput-specific) applications for a variety of professional and personal uses. For example, WLANs are being used in residential networks to support a wide range of applications such as remote controls, video from a security camera, delivery of video on demand, voice telephony, streaming audio, and Internet access. WLANs are also being used in community networks as a low-cost replacement for 3G broadband services by service providers. However, as is the case with all wireless networks, lack of bandwidth and interference constraints make WLANs a potential bottleneck. In order to support a variety of applications and to provide differentiated service quality, QoS mechanisms are required at the link or medium access control (MAC) layer of WLANs.

Several research efforts have tried to address the issues related to providing QoS in wide area wireless networks (e.g., [1]) or wireless ad hoc networks (e.g., [2]). The focus in the case of WANs is mostly on scheduling different classes of traffic over the air link, handling radio resources, and mobility management (handoffs and routing of traffic). In most other cases, the assumption is that QoS is handled at the IP or higher layers except in the case of scheduling algorithms where the wireless link bandwidth is shared among multiple stations according to some scheduling policy. In the case of ad hoc networks, there are many schemes that use higher layers for QoS support. In [2], a centralized bandwidth manager is used to allot bandwidth and manage admission control. WLANs do not have a complex backbone of mobile switching centers (MSCs), base station controllers (BSCs), or packet schedulers that can handle centralized allocation of resources. The predominant WLANs in use today and those expected to be in use in the future use decentralized or distributed medium access mechanisms based on carrier sensing, compared to centralized mechanisms that use time-division multiple access (TDMA)-like or reservation based access. Consequently, in WLANs, QoS provision needs to be distributed as well.

WLANs mostly use radio frequency (RF) transmissions for communications, although there are examples that employ infrared. There are a variety of medium access mechanisms and physical layers for WLANs specified by two major standards bodies: the IEEE in the United States and the European Telecommunications Standards Institute (ETSI) in Europe. IEEE 802.11 comes in many flavors: 802.11a, 802.11b, and 802.11g, supporting different physical layers but with the same MAC layer. The medium access mechanism was designed to be similar to the wired IEEE 802.3 (Ethernet) carrier-sense multiple access with collision detection (CSMA/CD) and is referred to as CSMA/CA (collision avoidance). The reason for collision avoidance in WLANs is that it is extremely difficult to detect collisions in the air, so a better approach is to prevent them to the extent possible. The ETSI standard is called HIPERLAN.
HIPERLAN/1 uses collision avoidance strategies similar to IEEE 802.11. HIPERLAN/2 uses a dynamic TDMA type of centralized medium access with no possibility of collisions. We focus our attention in this article on the widely deployed IEEE 802.11 WLANs and describe several proposed distributed mechanisms at the MAC layer for providing QoS support. Most QoS support mechanisms proposed for 802.11 use well known QoS techniques (e.g., priority assignment and fair scheduling) and map QoS metrics into some existing 802.11 MAC parameter, thus avoiding a redesign of the MAC protocol. We provide a taxonomy of the mechanisms and describe the essential concepts, problems, and advantages of each mechanism. From our study, we conclude that choosing the right set of MAC parameters as well as the QoS mechanism itself to provide predictable QoS in 802.11 networks is still an open problem.

The next section provides a brief background on IEEE 802.11 followed by a new taxonomy of distributed QoS schemes. Two sections contain descriptions of the QoS mechanisms used for the distributed modes of operation in IEEE 802.11. Concluding remarks are then provided, with a brief comparison of all of the schemes and directions for future research.

**Distributed MAC Protocols**

Centralized protocols, such as reservation TDMA or polling and scheduling schemes have received much attention from the research community, since they promise precise QoS guarantees. With centralized protocols, each mobile station (MS) requests the right to access the channel from a single point of coordination. The coordination point (called base station or access point) can perform admission control, bandwidth assignment, and channel access control. The major advantage of centralized protocols is that they can guarantee bandwidth resources (or deny admission). Examples of centralized protocols are the point coordination function (PCF) of IEEE 802.11 that employs polling, HIPERLAN/2 of ETSI, and numerous wireless asynchronous transfer mode (ATM) proposals. However, the adoption of these mechanisms has been limited due to high overhead, high cost/complexity, and issues in scalability, practicability, and flexibility. In contrast, distributed protocols are simple to implement and require smaller overhead. Although these protocols are currently not equipped with QoS support, they are being widely adopted. Therefore, the focus of this article is on distributed QoS mechanisms. There are several proposals for incorporating QoS mechanisms with distributed protocols. Ongoing work on a draft standard (IEEE 802.11e) is intended to provide QoS differentiation in WLANs in a distributed manner.

All distributed protocols are based on the principles of CSMA. Carrier sensing refers to an MS listening to the physical channel to detect any ongoing transmissions and backing off in case it detects any transmission. Although the CSMA family is simple to implement, the lack of synchronization between MSs and randomness in accessing the medium result in no guarantees for bandwidth resources or fair access. There are no built-in mechanisms to support priority, guaranteed delay bounds, or throughput. Examples of distributed protocols are the distributed coordination function (DCF) of IEEE 802.11 and the elimination-yield mechanism in HIPERLAN/1. We concentrate on 802.11 DCF networks in this article.

**The Distributed Coordination Function of IEEE 802.11**

The DCF of IEEE 802.11 is designed for data applications and is based on CSMA/CA. The channel contention procedure begins when an MS senses the channel to determine whether or not another MS is transmitting. The collision avoidance mechanism employs two techniques: interframe space insertion and a backoff algorithm. The interframe space (IFS) is the period of time an MS is required to wait after it senses an idle channel and enters the transmission process.

If the channel is idle for a period of time equal to the DCF IFS (DIFS), the MS can begin transmission. However, if the channel is busy, the transmission is deferred, as shown in Fig. 1. A backoff interval (BI) is randomly selected between a minimum contention window period (CW_min) and a maximum period (CW_max). The difference between CW_max and CW_min is the CW. A collision occurs if two or more MSs select

---

**Figure 1. The DCF mechanism.**
The DCF in IEEE 802.11 has advantages of simplicity, ease of implementation, and suitability for most data applications. However, DCF does not support QoS requirements or guarantee delay/throughput. Because of this, there is a need for QoS support mechanisms in DCF.

In distributed WLANs, QoS mapping and admission control have not been widely studied. Provisioning of network resources use two mechanisms, typically:

- Resource reservation
- Prioritization

In centralized protocols, distributed protocols employ prioritization where traffic is classified based on the application and resources are apportioned according to classes of priority depending on availability and demand. Traffic in a higher-priority class is given more resources than traffic in a low-priority class in a fair manner. The important QoS metrics for multimedia applications are delay, jitter, loss, and throughput. End-to-end delay is the time between the arrival of a packet and its successful delivery to the receiver. Another metric, access delay, is the time between packet arrival and packet transmission by the sender. Jitter is the variation of delay and is an important metric for multimedia applications. Finally, bandwidth is the measure of data transmission capacity and influences throughput, which is the amount of data successfully transmitted and received in unit time. Note that some of the data is lost in transit, and reducing the loss rate is an important QoS goal as well. Most of the QoS work in WLANs has concentrated on the throughput metric, and more work needs to be done in considering other QoS metrics as well. The term QoS enabling or QoS support mechanism will be used to describe a mechanism that can potentially provide some level of QoS.

Figure 2 contains a hierarchical taxonomy of distributed MAC mechanisms in IEEE 802.11 WLANs. At the highest level, MAC schemes can be categorized into distributed and centralized control protocols. In the class of distributed MAC protocols, as mentioned previously, we consider the DCF mode of 802.11. The 802.11 MAC protocol parameters, such as the IFS, CW, and BI have been suggested for QoS support. The approaches can be classified into priority- and fair-scheduling-based approaches, and are described in the next two sections.

**Priority-Based QoS Support**

The objective of various QoS support mechanisms proposed in the literature is to provide service differentiation by allowing faster access to the channel to traffic classes with higher priority. Faster access can be provided by allocating a smaller waiting time (IFS) or a smaller CW that results in a smaller BI on average. The IFS and BI values are deterministic once selected. The CW value corresponds to a range from which a random number is picked for backoff. Using the CW and selecting a random BI, however, introduces variations in delay and throughput. In either case, binding the priority to channel access makes these QoS support mechanisms unfair. As the number of MSs generating high-priority traffic increases, they tend to grab the channel, preventing fair access for low-priority traffic. The ongoing work of IEEE 802.11e [6] is proposing a modified combination of the

**TAXONOMY OF QoS MECHANISMS IN WLANS**

QoS is the ability of a network to provide some consistent level of ensuring data delivery over the network with different levels for different classes of traffic. A QoS system has several components, including QoS mapping, admission control, and resource allocation. QoS mapping refers to the translation of the QoS representations from one layer to the next. Admission control is used to determine whether a network is able to support the requested traffic with the requested network level QoS parameter. Resource allocation involves the allocation of suitable network resources according to the requested QoS.

\[
BI = \text{Random}(CW_{\text{min}}, CW_{\text{max}}) \times \text{SlotTime}
\]
approaches discussed below to provide throughput differentiation.

**USING THE INTERFRAME SPACE**

The idea behind QoS-enabling mechanisms that exploit the waiting time is to assign a smaller IFS value to higher-priority traffic. A higher-priority frame needs to wait for a shorter duration than a low-priority frame once the channel becomes idle and can seize the channel sooner. The low-priority frame finds the channel busy and has to either wait till the high-priority traffic has completed transmission or back off.

**Using Existing IFS Values for Priority** — Many researchers have proposed using IFS values that are already available from the 802.11 standard to differentiate between low- and high-priority traffic. In the 802.11 standard, the three different IFS values specified are the SIFS, PIFS, and DIFS. Deng et al. [7] suggest using the PIFS and DIFS values to differentiate between high- and low-priority traffic. Like DCF, this mechanism shows increases in average access delay and packet losses under high load conditions. However, the proposed mechanism can meet the bandwidth, delay, and loss requirements of high-priority traffic (video and voice) until the total offered load is very high (say, 0.9). Using only DIFS and PIFS, however, allows for only two priority classes. To differentiate between more than two priority classes, other alternatives must be explored. Deng has employed two different backoff algorithms after the IFS waiting time to support more traffic classes. However, this will introduce variations in throughput and delay due to the randomness of the backoff process. Another alternative is to use new IFS values, as discussed next.

**Using New IFS Values for Priority** — In contrast to using only the DIFS and PIFS values, the enhanced DCF (EDCF) proposals (presented to IEEE 802.11 working group E) introduce new IFS values [8, 9]. EDCF consists of up to eight prioritized queues that map onto and coincide with the standard 3-bit priority classes of 802.1p. A new type of IFS named arbitrary IFS (AIFS) is introduced. AIFS is an IFS value of arbitrary length. The AIFS value depends on the priority class of traffic, as shown in Fig. 3.

Each priority class has its own queue and backoff counter. Also, a small random time is added at the end of the IFS period to avoid collisions among frames in the same priority class. A potential problem of this mechanism is that the new AIFS values are longer than the existing DIFS. Therefore, the frame of an MS using the current DCF scheme receives higher priority than that of a QoS-aware MS using the EDCF mechanism. Aad et al. [10] also proposed using multiple IFSs to differentiate among priority classes.

**Discussion and Summary** — Although the proposed mechanisms can help differentiate the throughput for traffic with different classes of priority, fairness between different traffic classes is not guaranteed. Many schemes that are proposed with different IFS values are often combined with a backoff algorithm that could eliminate the priority provided by the IFS value. For instance, in current IEEE 802.11, DIFS is 50 µs and PIFS is 30 µs. Each slot in the CW is 20 µs. Suppose a frame from a high-priority traffic class waits for PIFS and enters backoff selecting six slots; it waits for 30 + 120 = 150 µs. A frame from a low-priority traffic class waits for DIFS and picks two slots for backoff. Thus, it waits for 50 + 40
A backoff algorithm is used after the waiting time (IFS) in DCF for preventing collisions, but it leads to randomness that cannot be controlled. Our simulations show that employing backoff leads to high variability of throughput and delay.

**Using Backoff Algorithms**

As described previously, the BI is an integer value that corresponds to the number of timeslots that a MS needs to wait after the IFS before it can transmit data. This section discusses proposals for QoS-enabling mechanisms that use a modified backoff algorithm.

**Contention Window Differentiation (CWD)** — The idea in the case of CWD is that given two classes of traffic, A and B, there are two ranges of the CW: CW_A (between CW_min,A and CW_max,A) and CW_B (between CW_min,B and CW_max,B). Since BI is a random number uniformly distributed between CW_min and CW_max, the two traffic classes are differentiated by the average BI values. These two CWs could, however, overlap.

Chesson et al. [8] and Benveniste [9] proposed mechanisms to modify the minimum and maximum value of the CWs. The values of CW are assigned such that the CW_min and CW_max values of low-priority frames are higher than those of high-priority frames (CW_min,i > CW_min,j and CW_max,i > CW_max,j, where priority of class i < priority of class j). The lower-priority frame selects a longer BI on average, whereas the higher-priority frame selects a smaller BI on average. Therefore, a higher-priority frame is likely to get access to the channel earlier than lower-priority frames. The differentiation of service depends on the amount of overlap between the CWs of different traffic classes. In the work by Barry et al. [11], the CW for a high-priority traffic class is between a CW_min of [8, 32] and a CW_max of 64. The CW for low-priority traffic is between a CW_min of [32, 128] and a CW_max of 1024. Because the overlap of the CWs of the lower- and higher-priority traffic is small, the delay between low- and high-priority traffic is clearly differentiated, as shown in their simulation results. A similar scheme, *distributed priority scheduling*, is proposed in [12] where the priority of every MS’s head-of-line packet is piggybacked onto RTS, CTS, data, and ACK frames. Using this information, MSs create a table of frames that are expected to be transmitted along with their priorities in a ranked list. Frames with the highest rank choose a smaller CW interval; those with lower rank have an additional waiting time and select the BI from a larger CW. The analysis and simulations in [12] show reduced mean end-to-end packet delays. In the extreme case, there is no overlap between the CWs of different traffic classes. This is discussed next.

**Contention Window Separation (CWS)** — As in the case of CWD, higher-priority traffic in CWS receives a CW that results in a smaller BI, whereas lower-priority traffic receives a CW that results in a longer BI. The CWs are completely separated, and the traffic from higher-priority classes is guaranteed to be transmitted before traffic from lower-priority classes. Low-priority traffic always has to wait longer than high-priority traffic, and it could face starvation.

An example of CWS is the algorithm proposed by Deng et al. [7] described by

\[ C_{\text{high}} = \left\lceil \text{rand\_uniform} \times 2^{i + 1} \right\rceil \]

\[ C_{\text{low}} = 2^{i + 1} + \left\lceil \text{rand\_uniform} \times 2^{i + 1} \right\rceil \]

Here \( i \) is the number of consecutive collisions. Although this mechanism completely separates the CW initially, the separation may not be valid in time because of the following. This mechanism is employed in each MS independently. As the number of consecutive collisions in each MS can be different, the CW of high-priority traffic \( C_{\text{high}} \) and that of low-priority traffic \( C_{\text{low}} \) can overlap and create an inconsistency among frames in the same priority class among MSs. Furthermore, it supports only up to two classes of priority. Therefore, Deng et al. combine this with the IFS mechanism described earlier to support more priority classes. Simulation results in [7] do show improvements in reducing delay and packet loss and increasing throughput for higher-priority traffic.

![Figure 3. The EDCF mechanism.](image-url)
Discussion and Summary — In DCF, after each consecutive collision, the value of CW is doubled:

\[ CW = 2^i - 1, \]  

where \( i \) is the number of consecutive times an MS attempts to send a frame and the initial \( CW_{\text{min}} \) is \( 2^3 = 8 \). As a result of this binary exponential backoff, the probability of waiting time or backoff time increases in direct proportion to the amount of time an MS has been waiting. This property is undesirable for time-sensitive traffic. Also, the BI selection is not uniformly distributed, but rather exponentially distributed where smaller BI values are more likely than longer ones. A long CW occurs only if multiple consecutive collisions occur. For example, BIs ranging from 0 to 7 time slots appear as choices for selection every time a new packet needs transmission. In contrast, a BI of 1023 timeslots appears as a choice only when the packet transmission has failed on eight consecutive trials. So an MS that has recently entered into contention could potentially transmit earlier than an MS that has faced several collisions.

A backoff algorithm is used after the waiting time (IFS) in DCF for preventing collisions, but it leads to uncontrollable randomness. Our simulations show that employing backoff leads to high variability of throughput and delay. Figure 4 shows the results from a simulation scenario with eight MSs in a 1 Mb/s IEEE 802.11 DCF WLAN, each having a sending rate of 100 kb/s. This figure demonstrates the high variability of throughput and access delay in IEEE 802.11 DCF. QoS enabling mechanisms using the backoff algorithms described earlier tend to develop inconsistencies in desired behavior over time.

QoS Support Using Fair Scheduling

In order to overcome the unfair apportioning of bandwidth created by binding the channel access to priority of traffic class, recently there have been proposals that use fair queuing mechanisms as part of the channel access. For instance, consider two traffic classes that need 200 and 100 kb/s, respectively. The mechanism is considered to be fair if the throughput levels for these classes are always in the ratio 2:1 on average. Fair scheduling algorithms [13] attempt to partition the network resource fairly among flows in proportion to a given flow weight. The idea here is to regulate the wait time so that traffic in each class has a fair opportunity to be sent, which is different from the schemes that bind channel access to priority. In this case, the bandwidth is fairly apportioned between different traffic classes. There are two such mechanisms proposed for 802.11 WLANs based on the backoff algorithms and one that uses both the IFSs.

Using Backoff Algorithms

The first technique, Distributed Weighted Fair Queuing (DWFQ), maps the traffic class into the CW value. The second technique, Distributed Fair Scheduling (DFS), keeps the CW fixed, but maps the traffic class into the BI within the CW.

Distributed Weighted Fair Queuing — Due to the fact that the length of CW is inversely proportional to the throughput, Banchs et al. [14, 15] proposed two modifications to the backoff algorithm. In the first algorithm, the CW is modified based on the difference between the experienced (actual) throughput and desirable (targeted) throughput. If the experienced throughput is smaller than the desirable throughput, the current sending rate is too low. The CW size is then decreased. In turn, this increases the priority of the MS compared to what it had previously. In contrast, if the network is either overloaded or the experienced throughput is higher than the desirable throughput, the CW value is increased to reduce the priority (and hence the data rate) of the MS. In the second algorithm [15], rather than comparing the experienced throughput to the desirable throughput, the authors apply a DWFQ algorithm (see several WFQ references.
We propose a distributed deficit round robin (DDRR) QoS-enabling mechanism based on the concept of deficit round robin — DRR scheduling at the MAC layer of IEEE 802.11. This mechanism makes use of fair queuing with the IFS value to enable QoS and it is of complexity $O(1)$ like DRR.

in [13]). All flows of all MSs are constrained to have the same ratio $(L_i)$ between throughput $(R_i)$ and a weight $(W_i)$ (i.e., $L_i = R_i/W_i$). By comparing its own $L_i$ to that of other MSs’ $L_j$, a given MS can adjust its CW accordingly, decreasing its CW if its $L_i$ is smaller than that of other MSs’ $L_j$ and increasing CW otherwise. The weights are used to differentiate between traffic classes and apportion the bandwidth between them. However, the randomness associated with using the CW remains, thereby increasing the variability of throughput and delay.

**Distributed Fair Scheduling** — Rather than having fixed ranges of CW for the low-priority frame and high-priority frame, Vaidya et al. [16] proposed DFS, based on fair queuing. Here, a packet with the smallest ratio between its length and weight receives an opportunity to transmit first. The weight represents a value associated with the throughput class. The weight of high throughput class traffic is larger than that of low throughput class traffic. The main idea of this mechanism is to pick a BI proportional to a finish tag. The finish tag is the ratio between the packet length and the weight of a frame given by $B_i = \left[ Scaling\_Factor \times L_i/\phi_i \times \rho_i \right]$. $B_i$ is the backoff interval, $L_i$ is the packet length, $\phi_i$ is the weight, and $\rho_i$ is a random variable uniformly distributed in [0.9, 1.1]. This random number with mean 1 is introduced to prevent a collision when two or more MSs count down to zero simultaneously. With the combination of weight and packet length in backoff calculation, traffic with different throughput classes can be treated differently. This mechanism is based on Self-Clocked Fair Queuing (SCFQ) [13, ref. 22] which has $O(\log(v))$ complexity where $v$ is the number of flows. Also, as the authors themselves note, the experienced throughputs are quite sensitive to the choice of frame lengths and weights, making it complicated to map the QoS requirement to the weight. As this scheme uses the BI parameter, it incurs the overhead associated with the waiting time for backoff.

**Distributed Deficit Round Robin**

We propose a distributed deficit round-robin (DDRR) QoS-enabling mechanism in [17] based on the concept of DRR scheduling [18] at the MAC layer of IEEE 802.11. This mechanism makes use of fair queuing with the IFS value to enable QoS and is of complexity $O(1)$ like DRR.

Traffic at each MS is categorized into classes with different QoS requirements. The throughput requirement of a traffic class determines its allotted service quantum rate (e.g., a traffic class requiring 100 kb/s gets service quanta at 100 kb/s). Each traffic class maintains a deficit counter of accumulated quanta and can transmit only when the deficit counter is positive. The deficit counter is reduced by the size of the transmitted frame. The deficit counter value is mapped to an appropriate IFS value (a larger deficit counter results in a smaller IFS value) as shown in Fig. 5. When an MS senses the medium as idle, it waits for this IFS time. If the medium is still idle, it will transmit its frame immediately. If the medium is busy, the MS waits for it to become idle, then waits for an additional time equal to the new IFS, then immediately transmits. In this basic scheme, we eliminate the backoff algorithm because of its contribution to fluctuations in throughput and delay. Instead, the IFS value resulting from a deficit counter is multiplied by a random number between 1 and a value $\beta > 1$ to reduce collisions between MSs with the same deficit counter. Preliminary investigations with the use of backoff in addition to IFS to further improve the performance of DDRR have proven to be effective under various conditions. We have found that the combination of the two MAC parameters (IFS and BI) enables DDRR to provide absolute as well as relative QoS levels. The choice of the right parameters and combination schemes need further study.

**Discussion of QoS Support Mechanisms Using Fair Scheduling**

We compare the throughput and access delay performance of the three fair scheduling schemes in Fig. 6. The results in Fig. 6 are based on simulations (using OPNET 9.0) with eight MSs with a sending rate of 100 kb/s in a 1 Mb/s WLAN. In the case of DDRR, each MS receives quanta at the rate of 100 kb/s. With DFS, the selection of weights is a difficult issue as noted in [16]. In our DFS simulations, we make the idealistic and best case assumption that the number of MSs in the WLAN is known, allowing the eight MSs in

---

**Figure 5. The DDRR mechanism.**
this example to be assigned the same weight (0.125), with all the weights summing to 1. With DDRR, the mapping between the quantum rate and the desired throughput is straightforward. Thus, the advantages of DDRR are its low complexity, easy mapping of the QoS metric (e.g., throughput to quantum rate), and good delay performance.

We see that DDRR and DFS have better performance than DWFQ in terms of reducing throughput variability. Also, the throughput variability of both DDRR and DFS is negligible compared to that experienced with only DCF in Fig. 4. DWFQ still demonstrates significant variability of throughput because of the randomness in the backoff algorithm. The results shown here consider a configuration without an access point and the RTS/CTS mechanism. Using the RTS/CTS mechanism with an access point can improve the performance of DWFQ [15]. Similarly, the delay variability is reduced for both DDRR and DFS. DWFQ has lower delay variability than DDRR or DFS. We have not included the performance of priority-based schemes (described earlier) in Fig. 6, as in this example there is only one traffic class, and the performance is similar to DCF with high throughput and delay variability.

**CONCLUDING REMARKS**

Research on distributed QoS support mechanisms for 802.11 networks is a relatively new area of study, motivated by emerging applications with QoS requirements. The proposed mechanisms use well-known QoS schemes (based on priority and fair queuing) from wired networks and map QoS requirements into 802.11 MAC parameters. However, the right set of MAC parameters and the appropriate QoS methods to employ are still open areas of research. Although we have presented some results on delay performance, most of the schemes thus far have focused on throughput guarantees. Other QoS metrics such as delay and jitter need more attention. Finally, the proposed mechanisms provide QoS differentiation, but no guarantees of QoS levels can be made without admission control and resource allocation, which is a fertile research area.

Some insights on QoS support mechanisms for WLANs are as follows. Most priority-based mechanisms cannot address fairness, and fair queuing schemes are necessary for this purpose. These distributed protocols have an overhead created by the waiting times (e.g., IFS and BI) used to differentiate between different traffic classes. Mechanisms based on the IFS parameter are the least complex as they only involve a simple precomputed waiting time that can be small and introduce small variability. The use of binary exponential backoff and a random BI value increases the variation in throughput and delay. The complexity of the overall scheme also depends on the choice of QoS mechanism. For example, different fair scheduling schemes have different computational complexity. Table 1 summarizes the comparison of the various distributed QoS support schemes on the basis of:

- **Ability to provide predictable QoS**
- **Overhead**
- **Complexity**
- **Fairness**

Fairness is determined by the correlation between the assigned priority (e.g., weight or quantum rate) and the experienced throughput, as defined earlier. The overhead is determined by the need for an additional field in the MAC protocol, the number of messages exchanged, and the average waiting time required by each frame before transmission. Complexity is determined by the computational requirement of the QoS support mechanism, practicality of the weight assignment, and the need for message exchanges. QoS metrics include the average

![Figure 6. a) Throughput and b) delay of 8 MSs using DDRR, DWFQ and DFS.](image-url)
aggregate throughput, throughput variability, and mean access delay.

Other comparison metrics not discussed here include power consumption related to the operation of the QoS support protocol and the ability of the QoS framework to adapt to the dynamism in the wireless channel characteristics, along the lines of the work in [11]. In all three WLAN standards, 802.11a, b, and g, multiple rates are supported (e.g., 1, 2, 5.5, and 11 Mb/s in 802.11b) that vary with the signal-to-noise ratio associated with an MS. Thus, the data rate accessible to an MS could dynamically change with time. Supporting QoS under heterogeneous and dynamic conditions needs further investigation.

**ACKNOWLEDGMENTS**

This work was partially supported by NSF Career Award ANI-9702389. We thank the anonymous reviewers for their valuable comments. We thank Jody Platt for his help in running simulations. We also express our gratitude to OPNET Technologies for providing the simulation tool for this study.

**REFERENCES**


**Biographies**

Wasan Pattara-Atikom (wapst7@pitt.edu) received his M.S. degree in telecommunications (2000) and M.B.A (2003) from the University of Pittsburgh. He is currently a Ph.D. student in the Telecommunications Program at the University of Pittsburgh. His current research interests are in QoS issues in wireless networks.

Prashant Krishnamurthy [M] (prashk@pitt.edu) is an assistant professor with the Telecommunications Program in the Department of Information Science and Telecommunications at the University of Pittsburgh, where he leads the development of the wireless information systems track for the M.S. in telecommunications curriculum. His research interests are in the areas of wireless data networks, wireless network security, and radio propagation modeling.

Sujata Banerjee [SM] (sujata@hpl.hp.com) is a senior research scientist at Hewlett-Packard Laboratories and an adjunct associate professor of telecommunications and computer science at the University of Pittsburgh. She holds a Ph.D. degree in electrical engineering from the University of Southern California in Los Angeles, and Bachelor’s and Master’s degrees in electrical engineering from the Indian Institute of Technology, Bombay. Her research interests are in quality of service issues in networked systems. She is a recipient of the National Science Foundation CAREER Award in Networking Research.