A Distributed Adaptive Algorithm for QoS In 802.11e Wireless Networks

Will Spearman
Clemson University, School of Computing
Clemson, SC 29632
wspearm@cs.clemson.edu

Jim Martin
Clemson University, School of Computing
Clemson, SC 29632
jim.martin@cs.clemson.edu

Keywords: Wireless MAC, 802.11, QoS, Simulation Modeling

ABSTRACT

Recent developments in the use of wireless networks have created a need for service prioritization for network traffic that is sensitive to throughput, latency, and jitter. The IEEE 802.11e standard provides enhancements that allow traffic with specific needs to be differentiated from normal traffic. While these enhancements have been shown to effectively improve latency and throughput for high priority traffic, they do not offer precise nor consistent control of performance levels. We present a method to dynamically optimize 802.11e contention parameters to provide more granular control over the network's quality of service (QoS). A distributed adaptive algorithm that extends 802.11e's Enhanced Distributed Channel Access (EDCA) is presented. We show that the enhancement provides more granular and consistent performance than that provided by the static algorithm used in standard 802.11e.

1.0 INTRODUCTION

There are a number of technical challenges presented by wireless networks due to the shared medium nature of wireless access. Among these are performance problems such as low throughput and high delays in networks with even a small number of active stations. As the number of nodes and amount of traffic increases on a wireless LAN, these issues become more serious. While quality of service (QoS) issues have been studied thoroughly in wired LANs, modern high-speed wired networks do not always require complicated QoS due to their relatively high bandwidths and error-free nature.

Applications that have specific bandwidth or latency requirements might not function well in a congested 802.11 network. Voice Over IP (VoIP) will experience reduced quality during a call when operating over a network experiencing high latencies or low throughput for the connection [8]. Likewise, applications using streaming audio and/or video will experience a reduction in quality. These problems can be more complex than just low throughput and high delay. Large variations in delay can also cause problems with protocols that attempt to adapt to networks with high delay, and in turn cause a considerable degradation in service quality. As wireless technology becomes ubiquitous, it is important to find solutions for these issues.

The IEEE 802.11e standard provides enhancements that allow traffic with specific needs to be differentiated from best effort traffic. While these enhancements have been shown to effectively improve latency and throughput for high priority traffic, they do not offer precise control of performance levels. We present a method to dynamically optimize 802.11e contention parameters to provide more granular control over the network's quality of service (QoS). A distributed adaptive algorithm that extends 802.11e's EDCA is presented. We show that the enhancement provides more granular and consistent performance than that provided by the static algorithm used in standard 802.11e.

This paper is organized as follows. We provide a background on 802.11 and then on QoS enhancements provided by 802.11e. Next we summarize related work. We describe our methodology, identify the problem statement, and describe the distributed algorithm. We show simulation results that demonstrate the benefits of our method. We conclude by identifying future work.

2.0 802.11 WIRELESS TECHNOLOGY

The 802.11 set of standards covers telecommunications and information exchange between systems in local and metropolitan area networks [1]. The 802.11 standard provides best effort packet services for the Medium Access Control (MAC) layer of wireless networks. This MAC layer provides wireless stations fair access to the medium in a best effort manner. In the following sub-sections we review the base 802.11 protocol.

2.1 The 802.11 MAC Layer

The original 802.11 MAC layer is built around two coordination functions that control medium access by the use of distributed coordination and centralized coordination. In the Distributed Coordination Function (DCF), the access control mechanisms are located at the station as opposed to the Point Coordination Function (PCF) in which control is centralized to the Access Point (AP). In 802.11 networks, DCF is always used, although PCF may be used optionally along-side DCF. Most 802.11 products support just DCF.

2.2 CSMA/CA In 802.11

In Ethernet networks, the primary method for medium access is CSMA/CD (collision detection) in which collisions are detected on the channel, and are handled by back-off counters that reduce future collisions by randomly increasing window sizes. Detection and recovery are efficient and feasible in wired networks due to the high bandwidth and low packet times of modern networks. This "reliable" nature of wired networks significantly reduces the impacts of collisions. In the wireless realm, however, interference can cause substantial noise resulting in frequently corrupted packets. Even more problematic is channel sensing is not possible since most radios can not simultaneously send and receive [1]. For these reasons collision detection is not possible for 802.11 wireless networks motivating the need for CSMA/CA (collision avoidance).

CSMA/CA works on the principle of listening before transmitting. By using wait times efficiently, this can lead to an algorithm by which all stations are allowed to gain access to the medium in a relatively fair manner and minimizing collisions using DCF or PCF, or both. This algorithm relies on inter-frame spacing to coordinate the communication of the stations. Four specific time intervals are defined as follows:

- SIFS (Shortest Inter-Frame Space): the wait time between the last transmission and high priority transmissions such as RTS and CTS frames and positive acknowledgments. Positive ACK frames are given priority so that currently communicating stations are given immediate feedback on the most recently sent frame. Since RTS and CTS frames are control frames, they are naturally given priority over other frame types.
- PIFS (PCF Inter-Frame Space): the minimum idle time for contention-free access such as PCF which is discussed in detail below. This interval allows the point coordinator priority over stations.
- DIFS (DCF Inter-Frame Space): the minimum idle time for contention based access such as DCF. Any station may claim the medium after this time interval.
- EIFS (Extended Inter-Frame Space): minimum wait time for a station that receives corrupted frames.

The frame spacing intervals allow DCF and PCF to interact seamlessly and with as few collisions as possible by always assuming the following relationship: slot time < SIFS < PIFS < DIFS < EIFS. The follow diagram shows the relationship of the inter-frame spaces to the access algorithms discussed in the next sections.

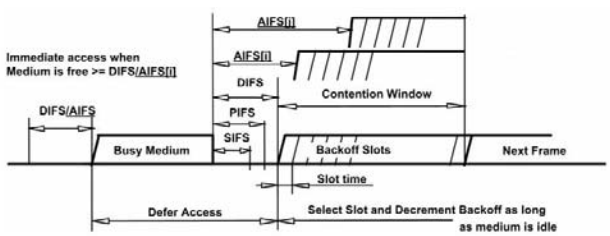

Figure 1. 802.11 transmission intervals.

In Figure 1 taken from [2], it is clear that components that are controlled by smaller time delay intervals will have a distinct advantage over those that use longer time intervals. In the following sections, it will be shown how these inter-frame spaces are the basis for providing control mechanisms with priority over stations, as well as providing station priority in the absence of centralized control.

2.3 DCF

DCF allows stations to transmit without a central coordinator. When a station wishes to transmit, and has sensed that the medium is free, it waits for a DIFS and transmits. If during the DIFS, the medium becomes busy, it begins decrementing a back-off counter, that is defined by the Contention Window (CW). The CW begins equal to CW_{min} and ends equal to CW_{max}. After each consecutive collision, the counter is set to a random value between 0 and CW. Each time a collision occurs the CW is increased until it equals CW_{max}.

If the CW reaches zero, and the medium is still free, then the station begins transmitting. If during the countdown, the medium is seized by another station, the station stops the counter and resumes after the transmission period. If the station senses the medium to be free, reaches a counter value of zero, and begins a transmission that results in a collision (no ACK received), the station will pick a new CW value.

DCF also includes an optional Request To Send/Clear To Send (RTS/CTS) mechanism to eliminate the hidden station problem. The hidden station problem occurs when two stations can sense transmissions of the AP, but not of each other. Due to their inability to receive each other's signals, the two stations can claim the medium simultaneously, and will cause a collision at a central destination. To prevent this, before sending a frame, a station transmits a RTS and then receives a CTS from the central station. Both of these frames include information regarding the time it will take to send the frames, which allows other stations to set a timer called the Network Allocation Vector (NAV) since the medium will be busy at least for that length of time. After that time, stations begin normal time interval waiting, and back-off counter decrementing. Since RTS/CTS frames are allowed to be transmitted after a SIFS, they have priority over normal DCF transmissions.

2.4 PCF

In PCF, medium access is controlled by a Point Coordinator (PC). The PC controls access by looking for stations wishing to transmit during a Contention Period (CP), and polling stations during a Contention Free Period (CFP). Together the CP and the CFP form a superframe which repeats for each time period. During the CFP, PCF is used to control access, then during the CP, DCF is used.

The CFP portion of the superframe begins with a beacon frame that contains management information such as protocol parameters and time synchronization. After the beacon frame has been transmitted, the CP polls stations in a round-robin manner, and upon successful response, allows the station to transmit either an ACK indicating it has nothing to send, or a DATA+ACK frame. Having received no response from a station the CP moves on, and the station is not allowed to transmit until the CP, or during the next CFP. The CFP ends when the time period specified by the beacon frame expires, or a CFP-EndFrame is sent. After the CFP has ended, a normal DCF period proceeds. However, since PIFS is shorter than DIFS, the PC can immediately seize the medium and begin another CFP if desired.

While PCF was intended to provide a form of QoS to 802.11 networks, it is generally agreed that it fails to provide this service adequately. Although PCF gets priority over DCF since the PIFS is always less than the DIFS, it suffers from the fact that individual network flows cannot be singled out for prioritization since the PC polls in a round-robin fashion. High priority can be given to individual stations, but affecting service on a more granular level is impossible with PCF. Also, polling can result in excessive overhead and large end-to-end delay when the number of stations is large [8].

3.0 QoS IN 802.11e

In an effort to give 802.11 networks true QoS, 802.11e was standardized. 802.11e introduced enhancements to the existing DCF and PCF, placed them under the heading of the Hybrid Coordination Function (HCF). The HCF is comprised of Enhanced Distributed Channel Access (EDCA), which is an enhanced DCF, and HCF Controlled Channel Access (HCCA), which has many traits in common with PCF. These two access methods work separately or together, just as in 802.11, where DCF is mandatory and PCF is optional. While the fundamentals of the original functions were not changed, augmented information allows HCF to provide QoS to specific flows and/or stations.

3.1 EDCA

In the EDCA, MAC layer parameters are used to provide priority to each traffic class (TC, also called Access Class) in a contention access manner similar to DCF. The parameters that can be manipulated are the Arbitration Inter-Frame Space (AIFS), the Transmission Opportunity Limit ($TxOp_{max}$), the CW_{min}, and the CW_{max}. These parameters are given default values at each station for each TC, or they can be overridden by an AP using special coordination frames.

Each parameter creates priority in a different way. The AIFS focuses on the time interval the TC must wait before trying to gain the TxOp. The $TxOp_{max}$ defines the length of time a station may transmit on behalf of a TC. The CW parameters prioritize by adjusting the back-off counter's minimum and maximum sizes respectively. Each parameter can be varied within TCs, while normal DCF rules apply between TCs within a station. For example, each TC is maintained in a separate queue within a station, and each queue contends for access to transmit on behalf of the station. If the back-off timer of two TCs within a station expire at the same time, the algorithm treats this as a virtual collision and will requires the lower priority flow to increment its collision counter and find a new back-off counter value.

In EDCA, the AIFS is a time interval that is equal to or greater than the DIFS such that higher priority stations can be given low values. When the AIFS expires, normal DCF operation for a station continues by decrementing the back-off timer. Therefore, TCs with low AIFS values will be more likely to gain access to the medium. Similarly, the CW_{min} value controls the size at which each TC starts its CW. A lower CW_{min} allows the flow to seek the medium sooner after the AIFS, and to be more likely to get the medium after a collision. When a collision does occur, each TC multiplies the current CW by the Persistence Factor (PF, usually equal to 2) to calculate the next back-off timer. The CW_{max} controls the maximum value to which a flow's CW can grow. A larger value will allow the flow to be less competitive during heavy collision, high load situations. Lower priority flows will have larger CW values and will wait longer when network traffic is causing many collisions. Finally, the $TxOp_{max}$ controls the length of time for which a station can transmit on behalf of each TC. Larger $TxOp_{max}$ values allow stations to send more frames during each use of the medium.

3.2 HCCA

In the HCCA portion of the HCF, a Hybrid Coordinator (HC) is used to centrally manage the medium in much the same way as PCF, with the exception that HCF uses parameterized QoS. Parameterized QoS refers to the use of specific information such as minimum data rate, maximum latency, etc that allows the HC to prioritize accordingly. When acting on behalf of a new TC, the station sends the HC the requirements of the TC. The stations transmit these requirements in the form of Traffic Specification (TSPEC) frames to the HC, which can accept or reject the request based on network conditions. Unlike PCF, the HC can poll stations during the CP, which allows the HC to provide TC prioritization as well as station prioritization.

Similar to the relationship between DCF and PCF, EDCA and HCCA are integrated within 802.11e yet operate independently. Even when HCCA is not used, EDCA uses superframes with CFPs and CPs. If a HC does not exist, or chooses not to poll during a superframe, the CP starts and EDCA rules are followed. The HC's transmission is governed by the PIFS time interval, and since this interval is smaller than the AIFS, the HC can obtain the medium before normal stations do. If the HC is active, it polls stations that have indicated a need to be scheduled, allows them to transmit, then ends the CFP with a CF-End control frame.

3.3 Other 802.11e Enhancements

The 802.11e standard defines both the HCF in order to provide specific flows and stations with high priority over others, but also defines other mechanisms to indirectly aid in this goal, such as: Contention Free Bursts (CFB), Special ACKs, and Direct Link Protocol (DLP).

Under normal operation, a station must contend for channel access after each TxOp. This can result in low throughput for data-rate sensitive applications even in the situation where it is given priority over other stations. Using CFB, if there is still time left in the TxOp after a frame is transmitted, the station is allowed to continue without waiting for a SIFS or the back-off counter. By using CBF in conjunction with higher $TxOp_{max}$ values, CFB can reduce the amount of overhead and allow a high priority station that already holds the medium to achieve greater throughput.

The second indirect improvement is the addition of two options to the QoS control field of data frames regarding ACKs. A station has the option to send packets with a flag set such that an ACK for that packet is not generated. In order to increase efficiency this NOACK can be used for applications where ACKS are not important, or do not signify any action on the part of the sender. Streaming media, for example, can tolerate packet loss but suffers greatly in the event of high latency. NOACKs in this situation improve efficiency by lowering the overhead required for the receiver. A second type of ACK called the BlockACK is also defined as aggregated ACKs that accumulate during a CFB. An ImmediateBlockACK can be requested by the sender after a CFB, and if necessary, the receiver can respond with a DelayedBlockACK if the receiver cannot respond before the sender's request timeout. These ACKs allow CFBs to be used effectively in an 802.11e environment, but do not directly improve prioritization.

Base 802.11 operation allows ad-hoc networking directly between stations, or infrastructure-based networking where stations cannot communicate directly. However in 802.11e DLP allows networks to use both simultaneously. When a station wishes to use DLP, it sends a DLP request to the AP. The AP forwards the request to the receiver. If the receiver supports DLP it will send a response through the AP back to the sender. The sender will then contact the receiver directly and begin the transmission. This direct communication can be especially useful when two stations are located near each other, but distantly from the AP. The signal between the two stations may be stronger and could result in fewer dropped frames. More importantly, eliminating the extra step of going through the AP can reduce the round trip times by half.

3.4 Current Status of 802.11e

As of late 2006 few network technology manufacturers have fully implemented 802.11e. Lack of push by consumers, and complexity of adding the standard to existing hardware has limited the proliferation of 802.11e in the marketplace [3]. In an effort to stimulate development towards full 802.11e deployment, the Wi-Fi Alliance (WFA) developed requirements for hardware to be Wi-Fi

Multimedia (WMM) compliant and Wi-Fi Scheduled Multimedia (WSM) compliant [3]. These two standards are subsets of 802.11e that allow stations to support applications that would benefit from prioritization. WMM uses EDCA while WSM uses HCF, but neither include the other enhancements discussed in the previous sections. WMM uses four categories in which to place traffic: voice, video, best effort, background. Each of the eight 802.11d categories are mapped evenly to a traffic class to allow backwards support for non-WMM stations. Any station sending data that is not assigned a traffic class, is considered best effort traffic. In this paper we refer to an 802.11b node (or just a 'b' node) as an 802.11e node configured to use the standard best effort traffic class. We refer to an 802.11e node (or just an 'e' node) as an 802.11e node that has at least one of its contention parameters configured to give the station priority over a best effort node.

4.0 PREVIOUS WORK

Previous work in the area of 802.11 QoS has shown that 802.11 network parameters can be adapted to provide better overall network service to general clients by maximizing throughput based on current network conditions [6,7]. It has also been shown that existing 802.11 design allows parameter tuning such that services like VoIP can be accommodated and given some QoS guarantees [8]. The fact that these approaches do not accommodate multiple priority levels or standardized parameters, has lead to the development of 802.11e. These developments, and their relevance as the basis of an adaptive algorithm, are discussed in the remainder of this section.

In [10], the importance of dynamic solutions to service prioritization becomes clear in the results presented. It is shown that the design of 802.11e increases the possibility of collisions and increases delay by adding an extra contention layer. In normal 802.11 networks the only collisions that can occur are those between stations as they try to gain access to the medium to transmit the packet at the head of the single send queue. In 802.11e, virtual collisions can occur when each AC queue must contend for access within the station's queue manager, as well as real collisions when the winning queue is allowed to transmit. The results in [10] show that throughput in 802.11e is decreased and latency is increased when compared to 802.11a networks due to this extra contention being introduced. Therefore, dynamic and aggressive tuning of the network parameters is required to achieve benefit from prioritization.

In [6,7] the authors evaluate a mechanism which allows dynamic tuning of the timing used in the back-off algorithm in 802.11. They showed that a dynamic algorithm based on the number of currently active stations, which manipulates the minimum back-off time can allow a wireless network to perform close to the theoretical capacity of the medium. The algorithm estimates the current performance and adjusts its back-off accordingly. Their findings show that static network parameters lead to under-utilization of the medium and show the importance of a dynamic algorithm. However, the network parameters are adjusted with global knowledge of the number of nodes in the network and collision rates.

In [9] the authors investigate two methods of choosing CW_{min}, based on proportional fairness and time-based fairness. They conclude that proportional fairness in a network based on weights provides higher throughput than time based fairness. Their work shows that CW_{min} can be tailored to a network to provide all nodes with fair access to the medium if priority mechanisms are used, and therefore it follows that the same principles can be used to provide nodes with unfair access, or differentiation, using contention window parameters.

In [4], the authors evaluate how a network with 802.11b and 802.11e nodes performs with different EDCA contention parameters, and how the delay and throughput are affected by these parameters. The 802.11b nodes model background traffic while the 802.11e nodes model high priority traffic. Four different contention parameters are tested: the initial size of the congestion window (CW_{min}), the maximum size of the congestion window (CW_{max}), the Arbitration Inter-Frame Spacing interval (AIFS), Persistence Factor (PF). These parameters can be adjusted to differentiate 802.11e traffic from 802.11b traffic present on the same network.

In [4], AIFS was shown to be the most effective contention parameter for protecting high priority traffic from background traffic. However, the authors show that using PF and CW_{min} for differentiation may have the advantage of allowing for better performance of the low priority traffic. The CW_{min} parameter can be characterized as a compromise between AIFS which is the most effective for e nodes, and no differentiation at all, which is the least adverse towards b nodes.

Similar work to that presented here is done in [11]. The authors use a two-level approach to fair, yet prioritized service. The first level of protection for high priority services guarantees that changing network conditions to not affect data streams such as VoIP and video that have constant QoS requirements. Using budgeted TxOp values for each queue, new flows are not allowed to have immediate access to their share of bandwidth regardless of their AC or parameter settings. This ensures that established flows are not disrupted by new flows. The second-level protection, called Fast-Backoff with Dynamic Adjustment when Fail or Successful, is the most similar due to its distributed nature. Under this scheme, when a station experiences a transmission failure, its CW is increased by a factor greater than 2 which results in a faster than exponential backoff. In addition to the CW increase, the station's CW_{min} is increased by a factor. When the station experiences a transmission success, CW_{min} is decreased by a factor, and the CW is reset to CW_{min}. This dynamic adjustment results in a dramatic decrease in the number of collisions, as well as more reliable service for priority data. A drawback of this method is that adjustments are made on all successes and all failures, and therefore best effort traffic has the potential to be inadvertently harmed when the high priority flows have sufficient QoS and continue to act aggressively.

Finally, as shown in [9], parameters can be tuned based on network conditions to allow better performance than a single setting. Although these settings are not changed dynamically, they do show that changing network conditions require changing parameters to use the channel efficiently.

5.0 METHODOLOGY AND PURPOSE

Figure 4 shows the simulated network model used in the research reported in this paper. The *ns2* simulator version 2.28 was used with an EDCA add-on [5] to simulate an 802.11e network. In Figure 4, the e nodes (labeled E, high priority) and b nodes (B, best effort) are connected to a base station (BS) which is bridged to a network of wired nodes (W). The wired nodes are connected with 100Mb/s connections while the wireless nodes use an 802.11b link speed of 11Mb/s. Each wireless station sends constant-bit-rate (CBR) UDP traffic (Source) to Wn (Dest) at a rate of 448Kb/s with a packet size of 210 bytes. RTS/CTS was disabled.

The EDCA model was extended to allow data collection. The main metric used to evaluate the performance of the system is the access delay, which is the time from a packet's entrance to the MAC queue to the time it is successfully sent. The mean access delay statistic is defined as:

meanAccessDelay = alpha * meanAccessDelay + (1-alpha) * accessDelay

where alpha = 0.875. Arriving packets were time stamped at queue entry and at packet transmission so that the average delays could be calculated. Additionally, end-to-end throughput and loss rates were measured.

The first set of experiments was designed to replicate the data in [4] which showed the effect each 802.11e contention parameter has on performance. Figures 2 and 3 demonstrate the results of these experiments. A equal number of e and b nodes were used in each run. The x-axis lists just the number of b nodes, but in actuality the total number of nodes involved in the simulation was twice this number. Figure 2 plots the mean access delay experienced by the e nodes, and Figure 3 plots the access delay experienced by the b nodes. The b nodes were configured using the TC2 (background) traffic class settings. The e nodes were also configured with TC2 settings except either the AIFS, CW_{min},, or CW_{max} parameter was set to a prioritized level specified by the TC0 class. Figures 2 and 3 visualize the individual impact that each contention parameter has on the access delay.

As found in [4], the results suggest that AIFS provides the best level of service for high priority traffic since in all cases it provides the lowest access times. In addition, CW_{max} provides the next best service while CW_{min} provides the least aggressive service. Figure 3 shows the effect of the number of nodes on the average delay of low priority (b) traffic for each parameter. AIFS differentiation causes a greater negative effect on the b nodes than the other contention parameters. CW_{max} only moderately affects b node performance, while CW_{min} is the least adverse to b access delays.

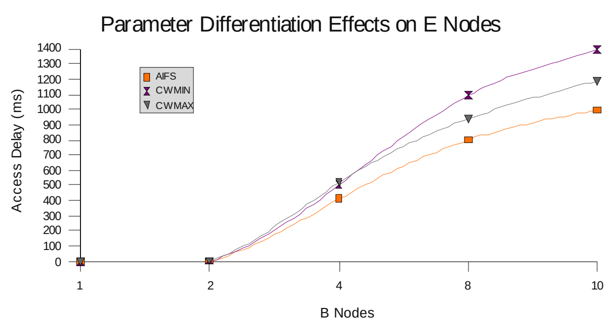

Figure 2. Mean access delay of e nodes as the number of nodes increase.

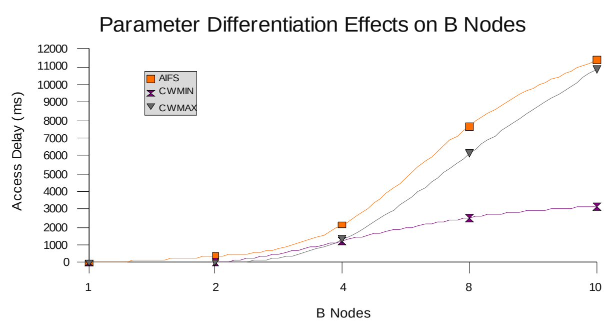

Figure 3. Mean access delay of b nodes as the number of nodes increase.

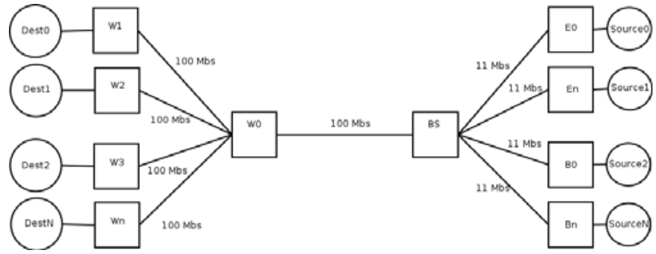

Figure 4. Simulation network structure.

A second set of baseline experiments were performed to further motivate the need for tighter QoS controls in 802.11e. Figure 5 shows the access delay and throughput of nodes that are identically configured with the default TC2 contention parameter settings. The line representing the access delay shows that the delay increases rapidly to over 2 seconds as the number of nodes increases. The majority of this delay is queuing delay experienced by packets. The line representing the throughput shows that the throughput for 1 node is 4 Mbps (the units in the figure are Kbytes per second) and drops to about 200,000 bps. Figure 6 shows the performance of a single e node using TC0 parameter settings that competes against one or more b nodes configured with TC2 settings. The figure plots the access delay of the single e node and the mean access delay of all the b nodes. Comparison of these two graphs show that the differences between TC0 and TC2 performance are rather extreme. The access delays of b nodes in Figure 6 are severely impacted by the single e node when compared to the access delays in Figure 5. When it is taken into consideration that the b node data of Figure 6 represents the average of many nodes, small changes reflect a large impact. This data is used in the rest of the experiments as motivation for finding a more linear way to provide performance than simply using TC0 or TC2. Simulations of TC1 are not shown due to the fact that performance is extremely similar to TC0 when using CBR traffic with CFB disabled.

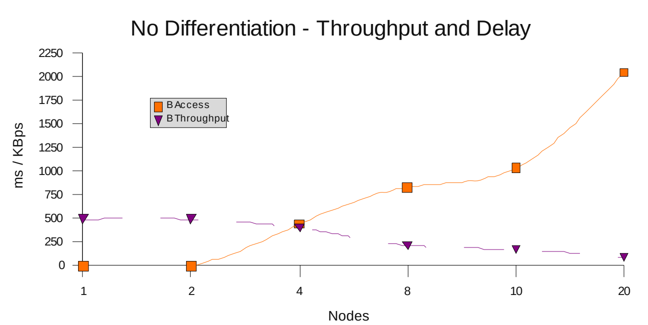

Figure 5. Access delay and throughput with identical node configuration.

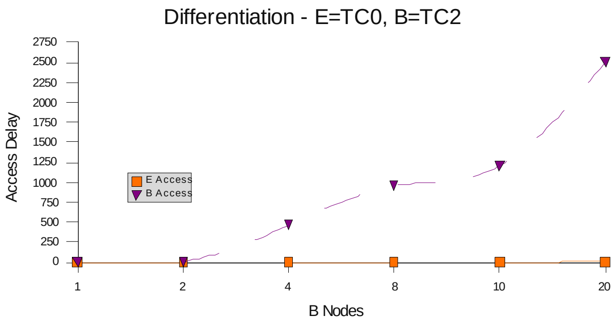

Figure 6. Access delay of e and b nodes.

Param/Priority	TC0	TC1	TC2	TC3
AIFS	2	2	3	7
Cwmin	7	15	31	31
Cwmax	15	31	1023	1023

Table 1. Parameters used in differentiation simulations.

5.1 Problem Statement

The simulations presented in the previous section demonstrate that:

- It is difficult to know the optimal value for the 802.11e contention parameters.
- The effective performance relative to different traffic classes depends on network dynamics.

These difficulties motivate our research. The performance difference between Figure 5 and Figure 6 shows that four static traffic categories may not be sufficient for QoS requirements that fall in between the absolute performance guarantee of TC0/TC1 and the best effort traffic class (TC2). Since TC1 performs similarly to TC0 the only choices for prioritization are using TC0, which negatively affects the b nodes, or TC2 which negatively affects e nodes. In the next section, we present a distributed adaptive algorithm that provides more consistent QoS control in 802.11e.

6.0 THE DISTRIBUTED ALGORITHM

The adaptive algorithm tunes the AIFS, CW_{min}, and CW_{max} parameters based on the performance of the network measured locally using a value we refer to as the aggressiveness ratio (*ar*). The *ar* is the ratio of the average access delay since the last *ar* calculation to an estimate of the minimal access delay. The *ar* value specifies a relative performance measure of how the station is performing compared to how well it could perform if there were no other channel contention. From this point forward, AR_PRESET refers to the target *ar* value set before the simulation for the high priority node, and *ar* refers to the aggressiveness ratio calculated during the algorithm operation. Figure 7 shows the pseudo-code that corresponds to the algorithm discussed in the rest of the section.

By dynamically adjusting contention parameters while the network is operating, the high priority nodes are able to maintain their quality of service under various levels of network load. This dynamic quality is important since it allows configurations for any delay requirements, rather than the narrow performance characteristics of TC0. The algorithm presented in Figure 7 uses a single setting called the AR_PRESET that can predictably provide performance on the continuum of access delays between TC0 and TC2.

One of the primary concerns for the algorithm is providing a way for each e node to derive stateful information about the system based on local observation. The algorithm uses mean access delay samples from before and after changes to the QoS values in order to measure success or failure of the parameter adjustment. After the change in access delay is assessed, the delay is saved, and a new time period begins. This feature allows the station to adapt quickly to current conditions.

```
E Nodes:
  BEGIN start_up_phase
     FOR i = random(0 to 50) to 50 by 1
        do nothing
     ENDFOR
     aifs = AIFS_OPTIMAL
     cwmin = CWMIN_OPTIMAL
     cwmax = CWMAX_OPTIMAL
     FOR counter = 0 to START_VAL by 1
        optimal_delay += average access delay
     ENDFOR
     aifs = AIFS_TC2
     cwmin = CWMIN_TC2
     cwmax = CWMAX_TC2
  END start_up_phase

  BEGIN normal_operation
     current_delay += average access delay
     IF (time_out < now)
        ar = current_delay / optimal_delay
        ar_ratio = ar / AR_PRESET
        IF (ar_ratio in range(0.8, 1.2)
           return
        ELSEIF (ar < AR_PRESET)
           IF (ar_ratio < 0.5)
              aggressive_decrease()
           ELSE
              relaxed_decrease()
           ENDIF
        ELSEIF (ar > AR_PRESET)
           IF (ar_ratio > 2.0)
              aggressive_increase()
           ELSE
              relaxed_increase()
           ENDIF
        ENDIF
     time_out = now + 0.50
     ENDIF
  END normal_operation
```

Figure 7. Distributed algorithm pseudo-code.

The second important design decision involves finding a reliable method of providing a node with knowledge of its overall goal for its access time, and more importantly, knowledge of when to give up pursuit of that goal. As a solution to this, a minimal mean access delay is measured when the node comes online. During start up, a node is given priority over all other nodes in order to estimate the minimum access delay to the base station. For this short period, the node lowers all parameters and assumes highest priority to get a "best case" measurement, to which it will then compare its mean access delays gathered later. Optimal delays reported by the algorithm remain reliable even as the network becomes overloaded since the station has absolute priority during the startup period.

A node's normal operation consists of increase and decrease adjustments to the EDCA values. The goal of the adjustments is to keep the *ar* in an acceptable range, which means the node is experiencing access delays that are near the minimum measured during start-up. Adjustments are made based on comparing the current *ar* value to the AR_PRESET for the station. Table 2 shows how the ar_ratio of the current *ar* to the AR_PRESET influences parameter changes. An "*a*" denotes an aggressive change while a "*r*" denotes a relaxed change. A minus denotes a decrease in priority while a plus denotes an increase in priority. The distinction between an aggressive change and a relaxed change is made by the order in which the parameters are changed. For example, according to Figure 2, an aggressive increase in priority would first try adjusting AIFS until exhausted, then try CW_{max}, then if the other two could not be adjusted it would try CW_{min}. The upper and lower bounds of adjustment are the static TC0 and TC2 parameters. A parameter is considered exhausted when it has been increased in priority to TC0 or decreased in priority to TC2.

Based on results of Figures 2 and 3, the algorithm changes each parameter by different units for aggressive changes. AIFS is incremented and decremented by one each time, while CW_{min} is changed by two and CW_{max} is changed by a factor of two. For relaxed changes, all parameters are changed by one.

ar ratio	< 0.5	0.5 – 0.8	0.8 – 1.2	1.2 – 2.0	> 2.0
Change	a-	r-	none	r+	a+

Table 2. Parameter Changes Made Based on the AR ratio.

7.0 RESULTS

Figures 8-11 were obtained using the adaptive algorithm outlined in Figure 7 using the simulation model illustrated in Figure 4. The workloads were the same as in the baseline simulation experiments. The graphs visualize simulation results that involve one high priority node and one or more best effort nodes. All access delays are measured in milliseconds. The e node implements the adaptive algorithm. The b nodes use the default TC2 parameter values with no adaptive behavior.

Figures 8, 9, and 10 show the results of single AR_PRESET values on various numbers of best effort nodes. An AR_PRESET of 1.5 as shown in Figure 8 performs similarly to TC0 settings, the results of which are shown in Figure 6. An AR_PRESET value of 1.5 means that acceptable access delays for high priority flows are less than 1.5 times higher than the measured optimal delay. Since additional nodes will create considerable contention for the high priority node, achieving an acceptable *ar* value requires the node to eventually adjust all its parameters to TC0. Therefore, a node with an AR_PRESET of 1.5 will perform similar to the static TC0 setting. In Figure 9 and 10, AR_PRESET values of 5 and 20 were used for simulations with varying number of b nodes similar to Figure 8. The results suggest that the adaptive algorithm is a mechanism that can provide more granular QoS control and that it can result in more consistent levels of performance between different traffic classes.

Figures 11 and 12 show the data resulting from a set of simulations similar to the previous, but with the AR_PRESET values varied. These data sets show that for any performance level there is an appropriate AR_PRESET than can be chosen.

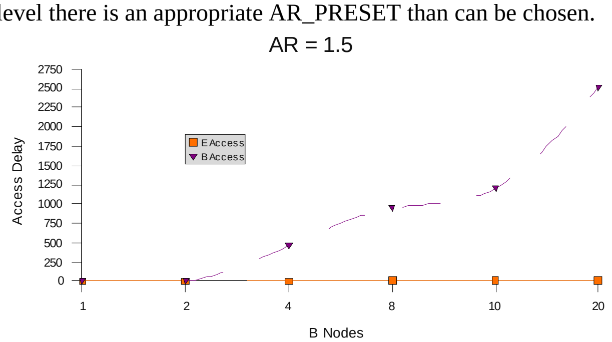

Figure 8 – Adaptive algorithm with AR = 1.5.

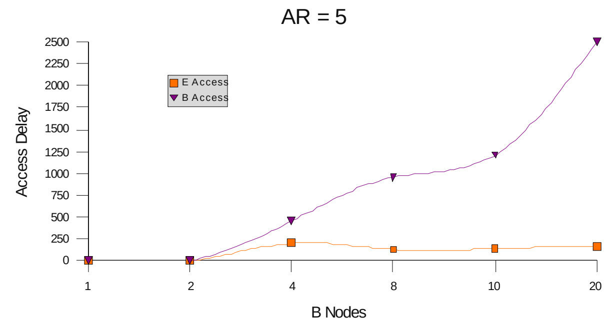

Figure 9. Adaptive algorithm with AR = 5.

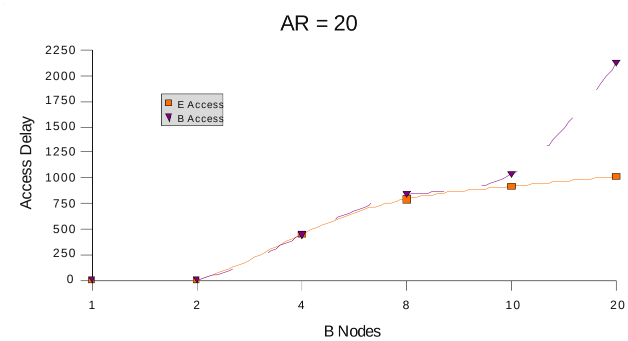

Figure 10. Adaptive algorithm with AR = 20.

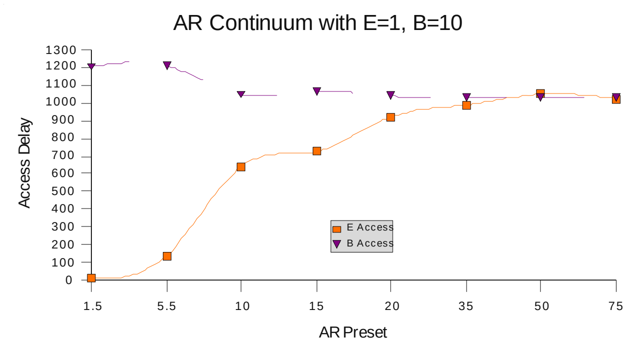

Figure 11. AR continuum for 10 b nodes.

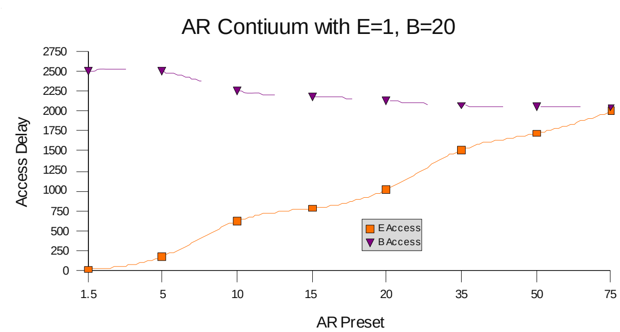

Figure 12. AR continuum for 20 b nodes.

8.0 DISCUSSION

The goal of the adaptive algorithm was to allow granular and consistent control of network performance in an 802.11e network. A specific QoS target can be achieved by choosing the appropriate AR_PRESET. Figures 11 and 12 show that even for heavily loaded networks, there is an AR_PRESET for any QoS level between the static TC0 and TC2 settings.

Although TC0 and TC1 can effectively isolate high priority traffic from best effort traffic, it might significantly (and unnecessarily) impact best effort traffic. High priority traffic does not always require absolute priority. Further, even high priority multimedia applications like VoIP have the ability to tolerate a certain level of latency and loss and still remain toll quality. Having a QoS mechanism that provides fine-grained control over network performance can be beneficial.

A further benefit of an adaptive algorithm is that it will use the channel more efficiently than static prioritization when utilization is low. For a station that does not require high priority, and whose current network conditions allow adequate performance, a high AR_PRESET means that prioritization only occurs when necessary to meet the target QoS. In other words, it will compete fairly with higher priority traffic until it reaches the threshold determined by the AR_PRESET.

9.0 CONCLUSIONS AND FUTURE WORK

We have presented an extension to 802.11e that dynamically adapts the contention parameters to meet a performance objective. While our algorithm is still under development, the ideas we present are a timely contribution to the community. Over the next 5 years, 802.11 is likely to face competition from 802.16 (WiMAX) primarily because QoS has always been a shortcoming of 802.11.

The work presented does have limitations, which we plan to address in future work. For example, our simulations involved unidirectional CBR traffic. In the future we will examine our algorithm using more realistic workloads. The algorithm currently uses the AR_PRESET to provide a relative performance objective. We are currently extending the algorithm to be application oriented so that the ultimate metric and goal is end-user VoIP quality, rather than MAC level access delays. Finally, we plan on validating the 802.11e simulation model by comparing simulation results with empirical results obtained from a similarly configured live network.

REFERENCES

1. ANSI/IEEE Std. 802.11b, 1999 Edition. IEEE Standard for Information technology – Telecommunications and Information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11. IEEE, 1999
2. ANSI/IEEE Std. 802.11e, 1999 Edition. IEEE Standard for Information technology – Telecommunications and Information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11, Amendment 8. IEEE, 1999
3. Wi-Fi CERTIFIED for WMM – Support for Multimedia Applications with Quality of Service in Wi-Fi Networks. Wi-Fi Alliance. September 1, 2004.
4. Swaminathan, Arvind and James Martin. 2006. "Fairness Issues in Hybrid 802.11b/e Networks". Consumer Communications and Networking Conference. 2006 3rd IEEE Vol. 1, Issue 8-10. 50-54.
5. Wiethölter, Sven and Hoene, Christian. 2003. Design and Verification of an IEEE 802.11e EDCA Simulation Model in ns-2.26. TKN Technical Report Series. Technical University Berlin. November, 2003.
6. Cali, F; M. Conti; E. Gregori. 2000, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit". IEEE/ACM Transactions On Networking, Vol. 8, No. 6, December 2000, 785-799.
7. Cali, F; M. Conti; E. Gregori. 2000, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism". IEEE Journal on Selected Areas in Communications. Vol. 18. No. 9 (September).
8. Chen, D; S. Garg; M. Kappes; K. Trivedi. 2002. "Supporting VBR VoIP Traffic in IEEE 802.11 WLAN in PCF Mode". Center for Advanced Computing and Communications, ECE Department. Duke University, Durham, NC. Avaya Labs Research, New Jersey.
9. Siris, Vasilios and Stamatakis, George. 2006. "Optimal CWmin Selection for Achieving Proportional Fairness in Multi-Rate 802.11e WLANs: Test-bed Implementation and Evaluation". International Conference on Mobile Computing and Networking: WiNTECH 06.
10. He, Dajiang and Charles Shen. 2003. "Simulation Study of IEEE 802.11e EDCA". Vehicular Technology Conference, Spring 2003. Vol. 1. 685-689.
11. Xiao, Yang; L. Haizhon; S. Choi. 2004. "Protection and Guarantee for Video and Voice Traffic in IEEE 802.11e Wireless LANs". INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communication Societies. Volume 3. Issue 7-11. 2152-2162.