Dynamic Adaptation Policies to Improve Quality of Service of Multimedia Applications in WLAN Networks *

Naomi Ramos, Debashis Panigrahi, and Sujit Dey
Department of Electrical and Computer Engineering
University of California, San Diego
{naomir, dpani, dey}@ece.ucsd.edu

Abstract

With the increased popularity of wireless broadband networks and the growing demand for multimedia applications, such as streaming video and teleconferencing, there is a need to support diverse multimedia services over the wireless medium. Recently pursued standardization efforts in IEEE 802.11e attempt to provide Quality of Service (QoS) differentiation mechanisms using two modes of medium access: polling-based and contention-based. However, there are a few limitations in the current approach with supporting inaccurate flow reservations, varying flow requirements, and congestion in contention-based access. In this paper, we address the above limitations by dynamically associating traffic flows appropriately to the two medium access modes and adjusting the duration of access in each mode. To show the effectiveness of our approach, we compare our adaptation policy with the 802.11e reference scheduler. We demonstrate that with our adaptation, the QoS of multimedia applications, in terms of delay and throughput metrics, can be significantly improved (2-4.5x).

1. Introduction

With the growing availability of multimedia content, there has been a significant increase in the use of multimedia applications, such as streaming video, teleconferencing, and voice over IP. Similarly, the recent affordability and improvements to support higher data rates have led to a wide spread adoption of Wireless LAN (WLAN) technologies. This can be seen in the greater use of WLAN networks in homes, offices, and commercial settings, such as airports and restaurants. With the popularity and technological advancements in these two areas, there will be a need to support multimedia applications over WLAN networks. However, the time-varying nature of wireless access and the diverse requirements of multimedia applications make the task of supporting wireless multimedia services challenging.

As a step towards meeting multimedia application requirements in WLAN networks, the 802.11 Working Group has been recently pursuing standardization efforts for a new standard, called IEEE 802.11e [13], that provides differentiation mechanisms at the Medium Access Control (MAC) layer. There are two categories of medium access in 802.11e: distributed and centralized. With the distributed channel access scheme, each flow gains access to the channel through a contention-based algorithm. This method is well suited for bursty traffic flows with unknown traffic requirements. With centralized access, a polling-based scheme is used to grant access to traffic flows. This scheme is well suited for flows that require guaranteed channel access and have predictable traffic. However, to be effective, the centralized controller requires accurate information about the flows prior to scheduling. The 802.11e standard also provides a framework called Hybrid Coordination Function (HCF) and a reference design [18] of a scheduler to multiplex between the two modes of medium access, and configure parameters of each mode appropriately.

Although 802.11e has provisions for supporting service differentiation, the reference HCF scheduler does have a few limitations. HCF assumes that real-time flows will reserve time for channel access during the centralized scheme. However, depending on the application, real-time flows may send reservation requests that are inaccurate and/or incomplete. Additionally, the reservation request only includes averaged values, such as mean packet size and required throughput; hence, HCF can only allocate a fixed polling schedule suitable for constant bit rate (CBR) traffic. However, there are multimedia traffic flows that do not have the CBR profile, such as quality-controlled MPEG4 or video-conferencing, and instead use Variable Bit Rate (VBR) encoding. Service providers commonly use VBR

---

[*] This research is supported by the University of California Discovery Grant and the Center for Wireless Communications (CWC), University of California, San Diego.

encoding of multimedia content to increase the capacity of the network by multiplexing between different VBR flows [15]. With possible inaccuracies in received reservations and traffic variations, HCF scheduling during the polling-based period can be inefficient and unsuitable for the traffic flows in the network, and can lead to unacceptable delays. Another limitation of HCF is that it restricts flows without reservations to contention-based medium access. Real-time traffic without *a-priori* knowledge of their flow information, as well as soft-real-time applications, such as web access, may also be mapped to the contention period. Though contention-based schemes have provisions to provide service differentiation to such flows under low network load, the achieved throughput can degrade drastically under high load leading to poor bandwidth utilization.

In this paper, we improve the support of QoS for multimedia applications by addressing the limitations identified above. We present an adaptation policy to dynamically associate traffic flows to the appropriate medium access mode. The proposed policy takes into account the possible inaccuracies in reservation information, the variance in flow generation and throughput requirements, and current system utilization. Additionally, the policy makes these adaptations with minimal effects on other flows in the network. To show the effectiveness of our approach, we compare our adaptation policy with the reference design of a scheduler for 802.11e[18]. We demonstrate that through our adaptation, we can achieve significant improvement in QoS in terms of delay and throughput metrics.

The rest of the paper is organized as follows. We first describe the 802.11e channel access schemes in detail in Section 2. Next, in Section 3, we motivate the limitations of the current approach of allocating flows to different modes of channel access. In Section 4, we present an algorithm to coordinate flows between two access modes. Experimental results follow next in Section 5, demonstrating the effectiveness of our adaptation policy. In Section 6, we discuss and contrast our work with other related research efforts. Finally, we conclude and describe our future directions.

2. IEEE 802.11e Background

In this section, we provide a brief description of our understanding of the enhancements being proposed in the IEEE 802.11e [13] draft to support service differentiation. The proposed standard defines a new operation mode called the Hybrid Coordination Function (HCF). As shown in Figure1, HCF multiplexes between two modes of medium access: a distributed contention-based approach guided by Enhanced Distributed Channel Access (EDCA), and a centralized polling-based approach called HCF Controlled Channel Access (HCCA).

2.1. Distributed Channel Access

In order to support service differentiation, 802.11e improves over the legacy 802.11 Distributed Coordination Function (DCF) by providing differentiated channel access called EDCA. Briefly, EDCA assumes that each flow can be categorized into different classes called *access categories* and using the traffic class information, it assigns different channel access parameters to prioritize medium access between different flows. The channel access parameters include Arbitration Interframe Spacing (*AIFS*), Transmission Opportunity (*TXOP*), and Contention Window parameters (CW_{min} and CW_{max}). The access parameters are decided by the Access Point (AP), and are beaconed to the nodes in the network.

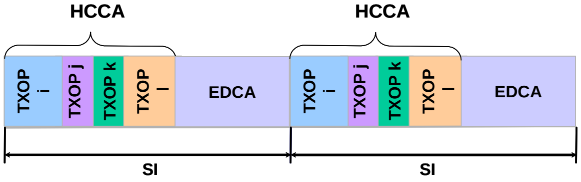


Figure 1. Hybrid Coordination Function (HCF) Channel Access

2.2. Controlled Channel Access

In addition to prioritized channel access in EDCA, the 802.11e protocol describes a centralized channel access scheme, called HCF Controlled Channel Access (HCCA), to provide guaranteed QoS. Figure 1 illustrates the channel access scheme used by 802.11e HCCA. Similar to the legacy Point Coordination Function (PCF), HCCA uses a polling-based mechanism and the medium access is controlled by the AP. The main difference between the legacy PCF and HCCA is the flexibility of when the contention-free period can occur. The AP can begin a contention-free HCCA period if the medium has remained idle for a PCF interframe space period, which is shorter than the minimum AIFS.

During the HCCA contention-free period, the AP polls nodes for a fixed time duration, called *TXOP*, which is computed based on reservation information sent to the AP by each of the flows. The *TXOP* is initiated by a poll request from the AP and during this period, transmissions can occur in both the uplink and downlink directions. This period allows for multiple contention-free transmissions and ends if one of the following conditions occur: neither the AP and the node have any packets left to transmit, the channel idle time has exceed the timeout period, or the time period expires. Note that the *TXOP* used in HCCA differs from that used in EDCA and is determined by the AP and calculated based on the flow requirements. The use of a fixed duration allows the AP to limit the time allocated to each node and is bounded by the default variable *dot11DefaultCPTXOPlimit*.

2.3. Hybrid Coordination Function Scheduling

Having briefly described the distributed and centralized channel access mechanisms, we now describe a reference scheduler [18] presented by the IEEE 802.11e Working Group. Nodes with strict QoS requirements send reservation requests containing flow information, such as mean data rate, mean packet size, MAC service data unit size (*MSDU*), and required service interval to the AP.

Using this reservation request, the scheduling policy decides the periodicity and the duration of the polls. The AP determines the minimum service interval (*SI*) to be used for all of the nodes, where the *SI* is the time duration between successive polls for the node. The selected *SI* is the highest submultiple of the 802.11e beacon interval duration while satisfying the service interval requirements of each flow; *i.e.* the selected *SI* should be less than the minimum of required service intervals of all flows. After deciding on the *SI* for the flows, the AP also allocates a fixed *TXOP* to each of the flow depending on the mean application data rate.

The maximum time spent in HCCA for each *SI* is limited by the *dot11CAPMax* variable, and the total controlled access time in a beacon interval is limited by *dot11CAPRate*. The above two variables limit the duration of controlled access period and bound the effect of controlled access mode on traffic flows in contention access mode.

3. Motivation

In this section, we motivate the limitations of the current HCF scheduling policy. In the reference scheduling policy, the AP maintains a clear separation between the centralized HCCA and distributed EDCA periods. With this scheduling, real-time flows with hard deadlines are restricted to being serviced for a fixed duration in the HCCA period, while other flows are left to contend in EDCA. Although this scheduling policy is simple, it has a few limitations in satisfying requirements of diverse applications. In the case of real-time flows, the HCCA's reference scheduler is not suited for Variable Bit Rate (VBR) traffic, such as quality-controlled MPEG4 video coding, high-motion real-time video coverage, video conferencing, *etc.* As described earlier, the reference scheduler allocates a fixed *TXOP* for each flow based on mean data rate, and each flow is serviced in fixed service intervals. Although this scheduling is well suited for Constant Bit Rate (CBR) traffic, variable traffic can cause high queue buildup and eventually lead to large delays and dropped packets. In [5], authors observe similar problems for VBR traffic. Another limitation of static HCCA scheduling is its inability to handle incorrect flow reservations. Although we mainly focus on limitations due to variable traffic, incorrect flow reservations lead to similar experienced delays.

To illustrate the limitation in handling VBR traffic, in Figure 2, we present the cumulative distribution function of the experienced delay of a CBR and VBR real-time flow in HCCA using the reference scheduler design. Note that if we consider a CBR traffic flow at 340 kb/s, the delay is minimal given the maximum tolerable delay of 40 ms. However, if the traffic of the flow is VBR with the mean of 340 kb/s with a standard deviation of 90 kb/s, the flow, with the time allocated in HCCA, is unable to handle the variation and leads to a packet delay that exceeds the maximum tolerable delay. Hence, to satisfy the VBR requirements, there is a need for allocation of additional time to VBR flows when needed. Using the algorithm presented in the next section, we demonstrate that dynamic allocation can reduce the delay experienced by VBR flows within the specified limits, as can be seen in Figure 7(a) in Section 5.

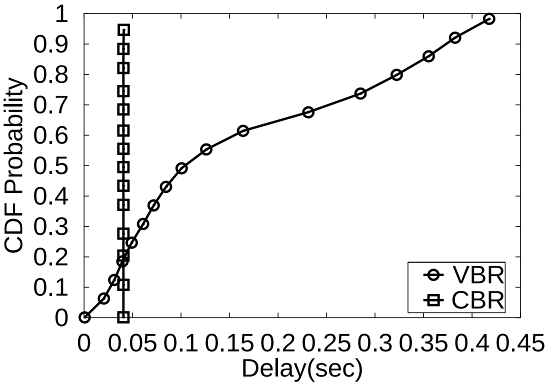


Figure 2. CDF Plot of Delay for HCCA Flow - CBR versus VBR

In addition to HCF's limitations with VBR traffic flows, another problem of HCF is the restriction of non-real-time traffic to the EDCA period. In the scenario where there are limited flows being serviced in HCCA, it is advantageous to schedule other flows in the network based on their priority and queue buildup. By using the time allocated in HCCA efficiently, we can improve the experienced delay and throughput of nodes in EDCA.

To illustrate this limitation, we consider a scenario of multiple flows, four pre-scheduled in HCCA and ten flows in EDCA. Figure 3 illustrates the throughput achieved by a prioritized EDCA flow with traffic characteristics of 400 kb/s. Note that despite its traffic class priority and available slack in HCCA, the EDCA flow experiences low throughput due to the high load in EDCA. However, by dynamically deciding to allow the EDCA flow to transmit during HCCA, based on the queue buildup and priority of the EDCA flow, we demonstrate that we can increase the EDCA flow throughput without affecting other flows in the network as can be seen in the results in Figure 8 in Section 5.

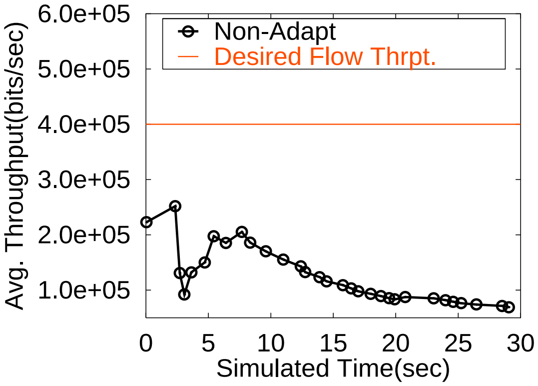


Figure 3. Average Throughput of EDCA Flow

4. Adaptation Policy

In the last section, we described the limitations of the 802.11e scheduling algorithm and the possible negative effects on Quality of Service (QoS). In order to prevent such problems, we propose a dynamic adaptation framework with the goal of improving QoS metrics, such as delay and throughput, for nodes scheduled in both HCCA and EDCA periods.

4.1. Overall Approach and Assumptions

The overall approach of our algorithm is to dynamically associate traffic flows to the two channel access modes and modify access privileges based on monitored traffic information. Specifically, with flows scheduled in HCCA, we consider tracking queue information to recognize when variations from the traffic reservation have led to a queue buildup. When this occurs, we allocate additional time to a traffic flow in order to reduce the flow's queue size and prevent a high experienced packet delay. If there is time available in HCCA, we allocate additional time by repolling a flow after all previously scheduled flows have been polled. However, if no time remains in the HCCA period, we attempt to decrease the queue buildup through better utilization of the EDCA period. The algorithm estimates the current load in the EDCA and depending on the load, we send a signal encouraging the HCCA traffic flow to send in the EDCA period. In addition to minimizing the delay of HCCA scheduled flows, our algorithm also attempts to improve QoS metrics for EDCA flows by allocating time in the HCCA period if there is time available. For this adaptation, we target high-priority flows that are suffering from throughput degradation due to congestion or channel variations.

Before describing our algorithm in greater detail, we first state our assumptions. We assume that the AP receives reservation requests, schedules the appropriate transmission opportunities to each node in the network, and polls all nodes using a fixed service interval, as defined by the reference schedule [18]. With this assumption, the ideal queue length for flows scheduled in HCCA after each *SI* is zero. Additionally, we assume that flows that have been scheduled in HCCA only send packets in the polling-based period and that the remaining flows without reservations are restricted to sending in the EDCA period. Finally, we assume that all flows provide queue length information to the AP using the control field of the 802.11e packet header.

4.2. Adaptation Algorithm

Having stated our assumptions, we now discuss our algorithm in greater detail. In order to modify channel access privileges and allocate additional time for flows, we answer the following questions: *(1) How do we recognize which flows need to be adapted and select between flows in a fair manner? (2) What are the acceptable conditions to run the adaptation and when should it be run? (3) How should we reallocate the time so as to minimize the effects on other flows in the network?*

For the first question, we determine which flows need to be adapted by comparing the queue length information sent in the packet header with an appropriate threshold. The queue threshold used for HCCA flows is dependent on the traffic requirements, current transmission opportunity, and service interval. The queue threshold for EDCA is larger relative to the HCCA queue threshold, since we want to allow EDCA flows to be scheduled in HCCA only when the queue buildup is significant. The main challenge is deciding how to make this selection in a fair manner. We sort the flows based on weights determined using timestamps of the last adaptation and current queue length information, and perform a priority-based selection.

Upon deciding which flows need to be adapted, the algorithm needs to decide whether it is possible and beneficial to allocate additional time and modify channel access privileges. For the HCCA flow allocation, there are two possibilities, additional allocation in HCCA or encouraged access in EDCA. With the first adaptation, there has to be sufficient remaining time in HCCA for the flow to be polled and send one packet. This is mainly done by comparing the remaining time in HCCA (computed using the *dot11CAPMax* and *dot11CAPRate* variables) to the time required to send one packet by the flow (using reservation request information). The greater challenge is determining when the second method should be used. If an HCCA flow is encouraged to send in EDCA when the network is congested, the traffic flow may suffer from retransmissions due to collision and cause negative effects on other traffic flows. Hence, to avoid such problems, the network load must be estimated by the AP. The network load is estimated using two statistics: *utilization ratio* and *collision count*. The AP calculates the utilization by monitoring the time used in EDCA, summing the transmission durations of successfully received packets over the total time available. Although the utilization ratio represents the network load under low load conditions, the value saturates as the load increases due to collisions. Under such conditions, we additionally use a collision metric that is determined by averaging the number of collisions that occur over the EDCA period.

Having selected the flows and deciding if/when the adaptation should occur, the final design aspect of the proposed algorithm is how to allocate additional time, so as to minimize the effects on other flows in the network. For allocation in the HCCA period for real-time flows, rather than extend the time duration for the flow, we allocate additional time by re-polling a flow after all previously scheduled flows have been polled. We use this approach in order to avoid unnecessarily causing delays for other flows scheduled in HCCA. In the case where HCCA allocation is not possible, the algorithm encourages flows to send during the EDCA period only if the network load is low. Rather than reduce the time allocation of other flows already scheduled in HCCA, this method gives the variable traffic flows an additional opportunity to send without negatively affecting the other HCCA flows. Additionally, the use of load estimation helps prevent negative effects on traffic flows in EDCA. Finally for the flows initially restricted in the EDCA period, the adaptation is only run when there is remaining time in the HCCA period, as given by the *dot11CAPMax* and *dot11MaxRate*. Additionally, since this algorithm is run in a round robin fashion, the algorithm attempts to fairly allocate time to the EDCA flows.

The pseudocode provided in Figure 4 and 5 summarizes the above algorithm. The first step of the algorithm is to monitor the current network conditions. The AP then proceeds with the original polling schedule and updates the queue statistics accordingly. Having completed polling the original schedule, the algorithm then allocates additional time in the HCCA scheduling period. By calculating the available time in HCCA (*hcca_slack*), the algorithm determines iteratively if there is available time to schedule in HCCA. In assigning flows to the additional time available in HCCA, we give priority to real-time flows already scheduled in HCCA over EDCA flows that need adaptation. The selection process of the flows is based on the weights described earlier in this section. In the case where the HCCA period is unable to support additional polling and there are remaining HCCA flows in need of additional scheduling, the AP explicitly signals to the real-time flow to send during the EDCA period.

```
/* Performed every service interval */
AP_schedule() {
  /*Monitor network conditions*/
  SCHED_EDCA_OK = monitor();
  /*Perform original polling schedule and update stats*/
  original_schedule_poll();
  /*Decide if additional nodes need to be polled*/
  coordinate_hcca_edca(SCHED_EDCA_OK);
}
```

Figure 4. Adaptation Pseudocode: Overall

4.3. Adaptation Framework

In order to run the described dynamic adaptation, we implement an adaptation framework consisting of two main functional components: *monitor* and *adaptor*, as shown in Figure 6. The monitor tracks the current conditions in both the HCCA and EDCA periods and provides the adaptor the flow weights and load information of the HCCA and EDCA periods, through a number of tracked statistics. Using these values and information given by the monitor, the adaptor decides which nodes can benefit from the switch, when nodes can be switched, and how they should be switched.

Having described the adaptation algorithm, in the next section, we compare our adaptation framework with the reference scheduler in order to evaluate the effectiveness of our approach.

5. Experimental Results

We now present the experimental evaluation of our adaptation framework over a diverse set of simulated scenarios with differing application requirements and network load. Using the metrics of delay, throughput, and fairness, we compare the performance of our algorithm with the reference scheduler [18]. Additionally, we present a preliminary comparison with the Flexible HCF (FHCF) algorithm, proposed in [5]. We begin by briefly describing the experimental setup and system configuration. Next, we evaluate the improvement in performance due to HCCA flow reallocation, EDCA flow reallocation, and the overall adaptation algorithm.

```
/*Evaluates hcca and edca to determine nodes to adapt*/
coordinate_hcca_edca(SCHED_EDCA_OK) {
  /*Determine leftover space, assume hcca_slack*/
  if (hcca_slack >= avgTXOP) {
    /*First deal with hcca and sort according to weights*/
    /*Returns list of flows needing to be scheduled*/
    sorted_hcca_list = sort_priorities(hcca_flows);
    /*We break if there is no more space and/or
      all the nodes have been satisfied*/
    while (hcca_slack > avgTXOP &&
          sorted_hcca_list != NULL) {
      /*Adds next node in sorted_hcca_list for polling*/
      schedule_hcca_flow(sorted_hcca_list);
      update_hcca_slack();
    }
  }
  /*We check to see if all hcca nodes are finished*/
  if (sorted_hcca_list != NULL) {
    /*Based on load, signal additional flows for EDCA*/
    if (SCHED_EDCA_OK)
      signal_EDCA(sorted_hcca_list);
  }
  /*Schedule edca flows into hcca if there is extra time*/
  if (hcca_slack > avgTXOP) {
    sorted_edca_list = sort_priorities(edca_flows);
    /*We break if there is no more space and/or
      all the nodes have been satisfied*/
    while (hcca_slack > avgTXOP &&
          sorted_edca_list != NULL) {
      /*Adds to polling list*/
      schedule_hcca_flow(sorted_edca_list);
      update_hcca_slack();
    }
  }
}
```

Figure 5. Adaptation Pseudocode: Coordination between HCCA and EDCA

5.1. Setup and System Configuration

For our experiments, we used the *Opnet* simulation environment [1]. In order to simulate the IEEE 802.11e standard, we modified the IEEE 802.11b MAC layer models to incorporate HCCA and EDCA service differentiation. We used the *Opnet* Direct Sequence Spread Spectrum (DSSS) physical layer with a data rate of 11 Mb/s. To generate different types of application traffic, we used OPNET application and profile configuration to set the packet size, as well as the inter-packet gap for the traffic generation process. The default characteristics of the simulated wireless networks are listed in Table 1. For the VBR video source model, we incorporated statistics from a trace of a MPEG-4 video stream of a video clip (VIVA-video) [2] with a mean data rate of 340 kb/s and a mean packet size of 1700 bytes. For the CBR video source model, we assume a 256 kb/s traffic stream with a frame rate of 25 frames/sec. Finally, for the EDCA data source, we use a source with a data rate of 400 kb/s. For these simulations, we assume a delay bound of 40 ms that is used to decide appropriate *SI* for scheduled flows in HCCA. In addition, we assume that each node is associated with only one flow and remains at fixed location that is distributed uniformly in the area of 300m x 300m, i.e. the range of an access point.

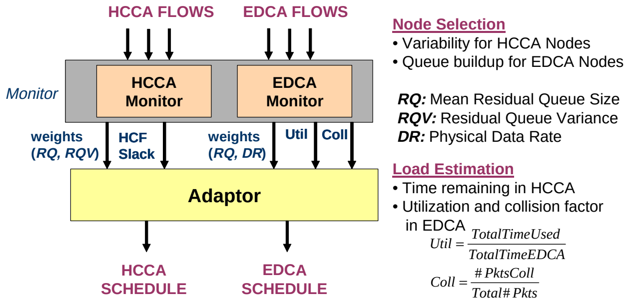


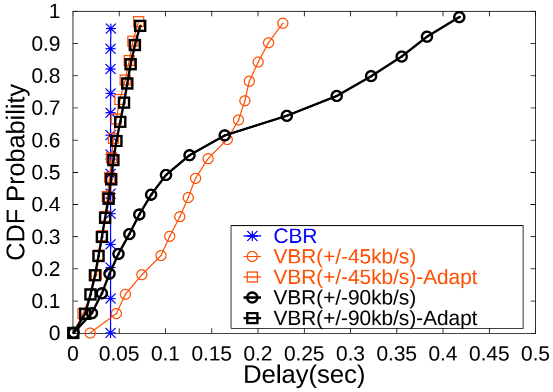
Figure 6. Adaptation Framework

Table 1. Simulation Parameter Settings

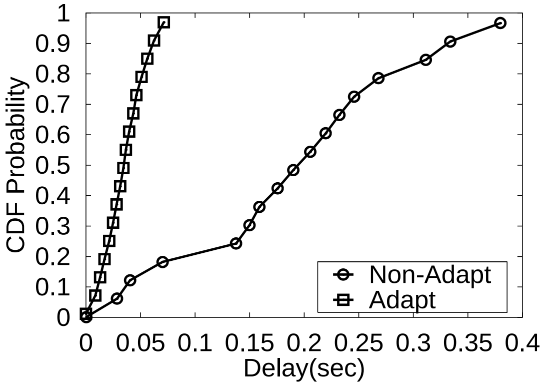
Parameter	Value
PHY Rate	11 Mb/s
Beacon Interval	100 ms
dot11DefaultCPTXOPLimit	2000 us
dot11CAPRate	21 us
dot11CAPMax	8000 us
SIFS	10us
DIFS	50us
PIFS	30us
SlotTime	20 us

5.2. HCCA Flow Reallocation

We begin by evaluating the performance of the algorithm on HCCA flows. Recall, the algorithm uses one of two methods to compensate for traffic variance. In the first technique, we allocate additional time to flows if there is remaining time in HCCA. For the first scenario, SCENARIO1, we use a topology consisting of one node with VBR traffic, three nodes with CBR traffic, and the other four nodes with bursty traffic. Figure 7(a) illustrates the cumulative distribution of the experienced delay of the VBR traffic with the same mean throughput but different variances. Note that depending on the variance in the traffic source, the variability in the experienced delay can be significant and can lead to an experienced delay exceeding the maximum delay of 40 ms. However, as seen from the graph, the



(a) Adaptation in HCCA



(b) Adaptation in EDCA

Figure 7. Effect of Adaptation on Delay for Variant HCCA Flows

proposed algorithm can significantly reduce the delay experienced by the HCCA flow. The observed reduction in delay in this scenario is due to the reallocation of the remaining time of the HCCA period to the VBR traffic flows. By reducing the delay of the VBR flow, the algorithm can avoid exceeding the maximum delay and decrease the packet loss experienced by the flow. In other words, the algorithm can improve the quality of the video traffic directly.

For the second approach, we consider a scenario, SCENARIO2, where the number of CBR flows has been increased to six nodes to saturate the HCCA time and the number of EDCA flows is increased from two to six. In this scenario, the HCCA period has no remaining time for reallocation. Since additional access allocation is not possible in HCCA, the proposed algorithm allows VBR traffic flows to send during the EDCA period if the load permits. Figure 7(b) compares the cumulative distribution function of the experienced delay with this adaptation in the case of four EDCA flows. For the simulation, the load estimation is performed using utilization and collision count thresholds fixed at 0.5 and 1.25 respectively, which were determined after experimental evaluations. As can be seen in the graph, the algorithm is able to outperform the reference scheduler significantly. The effectiveness of this portion of the algorithm is dependent on the current load of EDCA. In the scenario of six EDCA flows, the algorithm does not move the VBR flows as often and the mean delay after adaptation increases from 35 ms to 60 ms.

In addition to comparing our work with the 802.11e reference scheduler, we have done preliminary comparisons of the performance of our algorithm against the Flexible HCF (FHCF), proposed in [5]. Briefly, the FHCF algorithm monitors queue length and expands or shrinks the *TXOP* of each flow based on this information. In the case where the additional time is limited, the algorithm divides the time between the flows evenly. There are two main limitations with this approach: (1) By dividing the remaining time evenly, the expansion of *TXOP* duration may be insufficient in providing relief for residual queue overflow, and (2) Expansion of a flow's *TXOP* will lead to a delay variation for other flows scheduled in the HCCA period. For our preliminary comparison, we consider a scenario, SCENARIO3, with eight VBR flows scheduled in HCCA and two flows in EDCA. In this scenario, the remaining time in HCCA is insufficient to allocate additional time to all VBR flows. We compared the average delay of the VBR flows using our algorithm versus FHCF and observed that our algorithm achieved a smaller average delay than FHCF, 20 ms versus 50 ms. The improvement in delay is due to our algorithm's method of allocating additional time. Rather than divide the remaining time period between the flows, our algorithm allocates sufficient time for selected flows each *SI* to relieve their queue buildup. In order to insure fairness, we use a round robin approach to select a flow for additional time allocation.

5.3. EDCA Flow Reallocation

In order to evaluate the algorithm performance for EDCA flows, we consider a scenario, SCENARIO4, where the number of CBR traffic flows scheduled in HCCA is fixed at four and the number of EDCA traffic flows is fixed at ten. For this scenario, there is time remaining within the HCCA flow to be allocated to EDCA traffic flows. Due to high network load, the EDCA traffic flows can suffer from a queue buildup leading to a large decrease in throughput and increase in delay. Hence, EDCA flow allocation in the HCCA period is most effective under high EDCA load. Figure 8 shows the achieved throughput of a high-priority EDCA flow with and without adaptation. Note that our proposed EDCA adaptation is able to outperform the throughput achieved with the original HCF allocation.

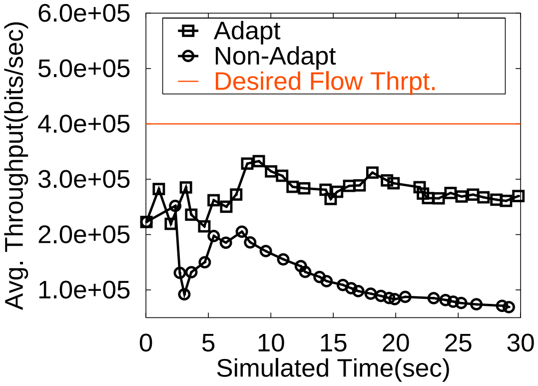


Figure 8. Average Throughput of EDCA Flow, With and Without Adaptation

5.4. Overall evaluation

Having investigated the performance of the two main components of the algorithm individually, we evaluate the overall effects when both flow allocation adaptations are running simultaneously. We consider a fixed number of CBR flows and vary the number of HCCA flows with VBR traffic (one to four) and the number of traffic flows in EDCA (four, six, eight). We consider a total number of twelve scenarios. Table 2 summarizes the results and averages the throughput for HCCA flows and is organized based on the number of HCCA flows considered. For example, in the case of two HCCA VBR flows, the average throughput of the two VBR flows in three scenarios with varying number of EDCA flows (four, six, eight flows) without adaptation is 159 kb/s. With the proposed adaptation, the achieved mean throughput is increased to 502 kb/s (an improvement of 3x). From this table, it can be seen that the improvement on the achieved throughput and delay using the proposed adaptation is significant, 2-4.5x and 3.5-6x respectively. In our results, we noticed that as the number of VBR flows is increased from one to four, there was an increase in the mean throughput achieved without adaptation. We believe that this is due to variations in the generated VBR traffic in Opnet simulation. However, it can be seen that with adaptation, the improvement in mean throughput is mainly dependent on the number of VBR flows and is reduced with increase in number of simultaneous VBR flows.

Table 2. Effect of Adaptation on Mean Throughput and Delay of HCCA VBR Flows

No. of VBR Flows	Thrpt (Kb/s)		Factor of Improv.	Delay (ms)		Factor of Improv.
	w/o	w/		w/o	w/	
1	119	536	4.5	173	39	4.4
2	159	502	3.2	145	40	3.6
3	164	500	3.0	240	40	6.0
4	230	465	2.0	171	49	3.5

Having looked at the improvement achieved by the HCCA flows, we turn to the EDCA flows and evaluate the effect of the algorithm on the achieved throughput. For the twelve scenarios described above, we observed that the queue buildup criteria for adaptation only occurs in the scenarios where eight EDCA flows are present. Table 3 shows the improvement of mean throughput of EDCA flows in these four scenarios. Note that we are able to improve the experienced throughput in the range of 15-40%. Furthermore, in order to ensure that the algorithm adapts fairly between the EDCA flows, we evaluate the Jain Fairness Index [14] in this scenario and observe that the algorithm has no negative effects on the fairness index and improves the fairness by up to 11%.

Table 3. Effect of Adaptation on Mean Throughput and Fairness Index of EDCA Flows

No. of VBR Flows	Thrpt (Kb/s)		Perc. Improv.	Fairness Index		Perc. Improv.
	w/o	w/		w/o	w/	
1	358	414	15.6%	0.89	0.99	10.8%
2	300	398	32.7%	0.96	0.99	3.2%
3	310	380	22.6%	0.98	0.99	1.0%
4	304	415	36.5%	0.97	0.99	2.0%

6. Related Work

There have been a number of schemes that have been proposed to support QoS over Wireless LAN. Since the legacy 802.11 system only provides best-effort traffic, initial research work investigated service differentiation by assigning different WLAN channel-access parameters to each of the traffic flows [4, 7]. Additionally, there has been some work concerning individual parameter adaptation to achieve higher throughput [19, 6]. All these attempts including many others were targeted towards providing QoS within legacy 802.11 and have led to the design of the new standard IEEE 802.11e.

Regarding IEEE 802.11e, the majority of published works have been mainly to model, simulate, and analyze the performance of IEEE 802.11e EDCA in comparison to the legacy IEEE 802.11 standard [17, 8, 16, 11, 12].

In addition to these performance evaluations, there has been some work evaluating possible adaptations in IEEE 802.11e. By changing channel access parameters of the contention-based EDCA access period, work in [21, 3, 20] propose dynamic adaptations to deal with network conditions, such as load and channel variance. For the centralized HCCA aspect of 802.11e, there has been limited work mainly focusing on the impact of using a different scheduler than the TGe-reference scheduler scheme, based on Weighted Round Robin (WRR) [18]. For example, [10] showed the improvement of throughput using an Earliest Deadline First (EDF) scheduler. Similarly, [22] proposed the static scheduled contention free burst (S-CFB) scheduling framework that minimizes the number of contention free bursts for energy-efficiency. As mentioned earlier, a closely related piece of work is presented in a recent paper, [5], where a scheduler is proposed to deal with the limitations of VBR traffic using the reference 802.11 scheduler. However, while they propose to change the transmission opportunity duration for a VBR flow, our algorithm attempts to minimize the negative effects on other flows by allocating time only after scheduled flows have been serviced and by choosing not to decrease other flow's transmission opportunities.

Finally, while there have been a number of techniques to improve aspects of EDCA and HCCA, there has been little literature on the coordination of the two periods. To the best of our knowledge, the interaction of centralized and distributed channel access has only been investigated in [9]. This work focuses on improving the aggregate throughput of access points by dynamically changing the ratio between the legacy IEEE 802.11 channel access periods, DCF and PCF. While this technique improves the aggregate throughput, the proposed adaptation in this paper looks at the advantage of dynamically changing which stations are in each channel access period. Note that the proposed technique directly uses monitored conditions to appropriately select the access method of a node. The proposed technique can complement the approaches of previous work mentioned above.

7. Conclusion and Future Work

In this paper, we identified the limitations of the 802.11e HCF scheduler in supporting inaccurate flow reservation, varying flow requirements, and congestion in contention-based access. We addressed the above limitations by developing an algorithm to dynamically associate traffic flows appropriately to the two medium access modes and adjusting the duration of access in each mode. In order to show the effectiveness of our approach, we compared our adaptation policy with the 802.11e reference design scheduler. We demonstrated that with our adaptation, the QoS of multimedia applications, in terms of delay and throughput metrics, can be significantly improved.

References

[1] Opnet simulation framework. `http://www.opnet.com`.

[2] Video traces for network performance evaluation. `http://trace.eas.asu.edu`.

[3] *Mobile Communications: 7th CDMA International Conference, CIC 2002*, volume 2524 of *Lecture Notes in Computer Science*, chapter Dynamic Offset Contention Window (DOCW) Algorithm for Wireles MAC in 802.11e Based Wireless Home Networks, pages 162–172. Springer-Verlag Heidelberg, Jan. 2003.

[4] I. Aad and C. Castelluccia. Differentiation mechanisms for IEEE 802.11. In *Proceedings of IEEE INFOCOM*, pages 209–18, Anchorage, AK, Apr. 2001.

[5] P. Ansel, Q. Ni, and T. Turletti. An efficient scheduling scheme for IEEE 802.11e. In *Proc. of WiOpt (Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Mar. 2004.

[6] A. Banchs and X. Perez. Providing throughput guarantees in IEEE 802.11 wireless lan. In *Proceedings of IEEE WCNC*, pages 130–8, Orlando, FL, Mar. 2002.

[7] M. Barry, A. T. Campbell, and A. Veres. Distributed control algorithms for service differentiation in wireless packet networks. In *Proceedings of IEEE INFOCOM*, pages 582–590, Anchorage, AK, Apr. 2001.

[8] S. Choi, J. del Prado, S. Shankar, and S. Mangold. IEEE 802.11e contention-based channel access (EDCF) performance evaluation. In *Proceedings of IEEE ICC*, Anchorage, Alaska, May 2003.

[9] X. J. Dong, M. Ergen, P. Varaiya, and A. Puri. Improving the aggregate throughput of access points in IEEE 802.11 wireless lans. In *IEEE Workshop on Wireless Local Networks*, Bonn, Germany, Oct. 2003.

[10] A. Grilo, M. Macedo, and M. Nunes. A scheduling algorithm for QoS support in IEEE 802.11e networks. *IEEE Wireless Communications*, 10(3):36–43, 2003.

[11] A. Grilo and M. Nunes. Peformance evaluation of IEEE 802.11e. In *Proceedings of IEEE PIMRC*, Lisboa, Portugal, Sept. 2002.

[12] D. Gu and J. Zhang. QoS enhancement in IEEE 802.11 wireless local area networks. In *Proceeding of World Wireless Congress (3G Wireless)*, 2003.

[13] IEEE-802.11WG. Draft supplement to standard for telecommunications and information exchange between systems - LAN/MAN specific requirements - part 11: MAC enhancements for quality of service (QoS). IEEE 802.11e Standard Draft/D5.0, Aug. 2003.

[14] R. Jain, G. Babic, B. Nagendra, and C. Lam. Fairness, call establishment latency and other performance metrics. Technical report, ATM Forum/96-1173, Aug. 1996.

[15] T. V. Lakshman, A. Ortega, and A. R. Reibman. VBR video: Tradeoffs and potentials. In *Proceedings of IEEE*, volume 86, pages 952–973. May 1998.

[16] A. Lindgren, A. Almquist, and O. Schelén. Quality of Service schemes for IEEE 802.11 wireless LANs - an evaluation. *In Special Issue of the Journal on Special Topics in Mobile Networking and Applications (MONET) on Performance Evaluation of Qos Architectures in Mobile Networks*, 8(3):223–235, June 2003.

[17] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. IEEE 802.11e wireless lan for quality of service (invited paper). In *Proceedings of the European Wireless*, volume 1, pages 32–39, Florence, Italy, Feb. 2002.

[18] J. Prado. Mandatory TSPEC parameters and reference design of a simple scheduler. IEEE 802.11-02/705ar0, Nov. 2002.

[19] D. Qiao and K. G. Shin. UMAV: A simple enhancement to the IEEE 802.11 DCF. In *Hawaii International Conference on System Sciences*, Hawaii, Hawaii, Jan. 2003.

[20] N. Ramos, D. Panigrahi, and S. Dey. ChaPLeT: Channel-dependent packet level tuning for service differentiation in IEEE 802.11e. In *Proceeding of Intl. Symposium on Wireless Personal Multimedia Communications*, pages 86–90, Yokusuka, Japan, Sept. 2003.

[21] L. Romdhani, Q. Ni, and T. Turletti. Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad hoc networks. In *Proceedings of IEEE WCNC*, Mar. 2003.

[22] L. Zhang, Y. Ge, and J. Hou. Energy-efficient real-time scheduling in IEEE wireless LANs. In *IEEE International Conference on Distributed Computing Systems*, Providence, Rhode Island, May 2003.