A Controlled-Access Scheduling Mechanism for QoS Provisioning in IEEE 802.11e Wireless LANs

Yaser Pourmohammadi Fallah
Dept. of Electrical Engineering
University of British Columbia, Canada
yasserp @ece.ubc.ca

Hussein Alnuweiri
Dept. of Electrical Engineering
University of British Columbia, Canada
hussein @ece.ubc.ca

ABSTRACT

Wireless Local Area Networks (WLAN) are being deployed at a rapid pace and in different environments. As a result, the demand for supporting a diverse range of applications over wireless access networks is becoming increasingly important. In particular, multimedia applications, such as Video and Voice, have specific delay and bandwidth requirements that cannot be fulfilled by the current IEEE 802.11-based WLANs. To overcome this issue, new enhancements are being introduced to the Medium Access Control (MAC) layer of the 802.11 standard under the framework of the IEEE 802.11e standard which is still a work in progress. The 802.11e standard offers new features for supporting Quality of Service (QoS) in the MAC layer, it however does not mandate a final solution for QoS issues and intentionally leaves it to the implementers to devise their own methods using the available features. We present a solution that employs the controlled access features of the 802.11e to provide per-session guaranteed quality-of-service. Our design comprises of a scheduler that assign guaranteed service times to individual sessions using a fair scheduling algorithm. We show that the proposed solution outperforms other methods that are contention and priority based.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - *Wireless Communication*

General Terms

Algorithms, Design

Keywords

Wireless LAN, 802.11e, QoS, Multimedia over WLAN, WFQ, Virtual Packet Scheduling

1. INTRODUCTION

Broadband wireless services are becoming increasingly available throughout the world. The availability of broadband services gradually changes consumer habits and encourages the use of demanding applications such as voice and video streaming. In fact, to many consumers personal computers and handheld devices are being viewed more and more as complex multimedia boxes rather than powerful computing machines. As a result, it has become inevitable and vital for the future growth of wireless access networks to provide the Quality of Service (QoS) required by multimedia applications such as voice and video conferencing and streaming.

The landscape for access network development has recently changed with the advent of affordable and easy to setup wireless Local Area Networks (WLAN). In particular the IEEE 802.11 standard [1] paved the way for wide scale deployment of wireless hot spots and LANs. Many hotels, residential areas and most enterprises and university campuses have already deployed an 802.11 based wireless access network. However, the 802.11 standard has not been initially designed for demanding applications such as multimedia. In presence of heavy traffic load, multimedia applications do not perform well and become almost unusable for users. This fact along with the need for more advanced features in the Medium Access Control (MAC) layer of the 802.11 standard motivated the development of the 802.11e standard [2] (still in progress and at draft stage at the time of this writing). The IEEE 802.11e standard provides an enhanced MAC that can be used with all physical layer (PHY) technologies that are described under the 802.11 standards (802.11a/b/g, etc.).

The need for providing Quality of Service (QoS) for real-time applications in 802.11 networks has been driving research activities and standardization efforts for some time [5]. Current research results offer some mechanisms to provide basic levels of QoS differentiation to aggregate flows mainly in the form of priority services (e.g. [5][6][7]). However, very little work has been dedicated to providing per-session guarantees in WLANs, a necessary feature for most multimedia applications such as video and voice.

The 802.11e provides adequate mechanisms and capabilities that can be used to provide different levels of QoS. These features are described under the two general frameworks for contention based and controlled based access. Most of the research in the past has been dedicated to studying the contention based mechanisms while controlled based methods that in our opinion are better suited for providing QoS are not studied as much [5].

In this article we focus on the enhanced controlled access features that are described in the 802.11e standard draft. It must be noted that the standard, intentionally, only provides the mechanisms and features and not a mandatory final solution. This is done to let the vendors devise their own proprietary solution while still conforming to the standard. In fact, the task of devising a QoS solution is left open in the standard and it is the responsibility of vendors or implementers to devise a complete solution. Our proposal in this article fills this gap and utilizes the

Q2SWinet'05, October 13, 2005, Montreal, Quebec, Canada.

available 802.11e features to achieve the desired guaranteed per-flow services for Multimedia applications. It must be noted that the MAC layer services in any wireless environment are always conditioned on the availability of service from the physical layer. This is also the case for our proposed solution as we only target the MAC layer in this article. The 802.11 standard provides features such as reduced and multi rate operation to cope with the impairment in the channel condition. These features can be easily utilized in our proposed MAC solutions.

The proposed solution provides guaranteed services to flows that make reservation with the WLAN Access Point (AP) by means of such protocols as Session Initiation Protocol (SIP) [4], while at the same time, allowing the normal contention based access to take place to use the remaining capacity of the channel. This approach is different from the de-facto solutions in which long alternating contention free and contention periods were generated, resulting in uncontrolled delay bounds and inefficient operation. The design, described in this document is called Controlled Access Phase Scheduling (CAPS). The CAPS algorithm is based on the ideas of Virtual Packet and mixed scheduling of uplink and downlink flows [3], but also adds a new queuing framework and a new scheduling discipline.

In this article, we first briefly overview the MAC layer of the 802.11 standard and the new features that are provided in the 802.11e standard draft. In particular we highlight the controlled access mechanisms. We will then present our new access scheduling framework that is designed especially for the 802.11e MAC, and capable of providing per-session QoS guarantees for voice and multimedia applications (given ideal channel conditions). The performance of the proposed solution is compared with other existing mechanisms.

1.1 Wireless LAN Standard: IEEE 802.11

The 802.11 standard specifies the MAC and Physical layers of wireless Local Area Networks. The part we attend to in this article is the MAC layer. The standard offers two modes of operation: Infrastructure, which is centrally controlled by an Access Point (AP), and Ad-Hoc, which is distributed. The discussion in this article is applicable to the infrastructure mode.

The basis for 802.11 MAC is a CSMA/CA mechanism (Carrier Sense Multiple Access with Collision Avoidance). Carrier sensing is done through physical sensing of RF carrier as well as a virtual carrier sensing in the MAC itself. Collision avoidance in 802.11 MAC is performed by a mechanism called Distributed Coordination Function (DCF). DCF specifies the rules for accessing the wireless medium in either a contention based or controlled manner. A station can access the medium in a Contention Period (CP) in which any station may independently try to access the medium according to DCF rules. It can also use a Contention Free Period (CFP), in which the stations are only allowed to respond to poll messages sent by an Access Point (AP) (refer to [1] for detailed information).

DCF uses inter-frame space (IFS) time intervals to coordinate channel access amongst stations (STA). Each station that has a frame to send is allowed to access the channel if it finds the medium idle for longer than a predetermined IFS time. The IFS is different for each type of frames, data frames use 'DIFS' time, AP uses a shorter PIFS time for sending a polling message or beacon; while Acknowledgment (ACK), Clear To Send (CTS), and response to poll messages use 'SIFS' time which is the shortest amongst IFSs. This shorter waiting time gives them higher medium access priority than the other frames (Figure 1).

Stations that find the medium busy perform a back-off procedure by waiting for a random number of slots before they try to access the channel again. The random number is uniformly chosen from a contention window (CW). The backoff counter is decremented after the elapse of each idle slot. The counter decrement is suspended whenever channel becomes busy. When the counter expires the station performs a deferment for DIFS again and then if the channel is still idle accesses the channel and transmits its frame. If the transmission is successful the contention window size is reset to its default CWmin; but if the transmission fails (no acknowledgement is received) the station must double the size of its contention window and then select a random number for backoff counter from the new contention window. This procedure reduces the probability of collision happening again. The details of DCF and 802.11 MAC operation are found in [1].

While DCF is a mandatory function in MAC, an optional mechanism, called Point Coordination Function (PCF), has also been defined in the standard. PCF resides in the access point and provides a contention free access method by polling individual stations whenever it wants to send to or receive data from them. Contention free periods (CFP), controlled by PCF, are repeated periodically. When the WLAN is controlled by an access point (AP), a beacon is also sent on a periodic basis. If PCF is supported, this beacon becomes the signal that indicates the start and end of a CFP.

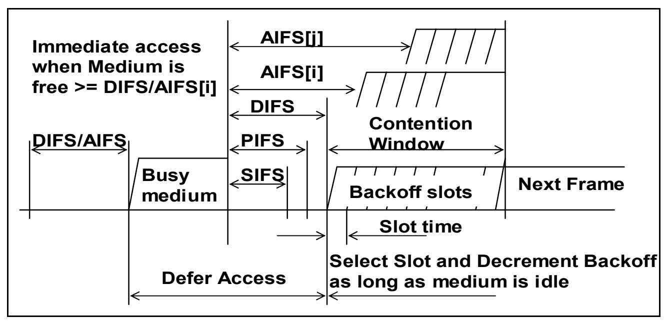

Figure 1 Some IFS relationships in DCF and EDCA

1.2 QoS Enhancements: IEEE 802.11e

The shortcomings of the IEEE 802.11 standard in supporting QoS are addressed in the upcoming IEEE 802.11e standard [2]. The 802.11e introduces new frame formats with QoS information fields, the capability to poll a station even during the contention period and new concepts such as transmission opportunity (TXOP). It also enables differentiation between different classes of traffic through the use of different contention window and IFS waiting times. The basic building block of MAC in 802.11e is again DCF, but with some modifications. Accessing the medium is controlled by the Hybrid Coordination Function (HCF) protocol. HCF offers two access mechanisms; EDCA (Enhanced Distributed Channel Access) which is an enhanced DCF is used for contention based access, and HCCA (HCF Controlled Channel Access) that replaces PCF is used for controlled access. 802.11e defines 8 different traffic priorities in 4 access categories. It also enables the use of traffic flow IDs, which allow per flow resource reservation.

Under EDCA access mechanism, depending on the type of a frame (Data or Control) and its priority, different IFS values are used (Arbitration IFS or AIFS in Figure 1). The backoff windows are also different for each priority. Shorter AIFS times and smaller contention windows give higher access priority. This prioritization enables a relative and per-class (or aggregate) QoS in the MAC. The 802.11e standard allows for dynamically adjusting most EDCA parameters, facilitating performance enhancement using adaptive algorithms [10].

Under HCCA, access to the medium is controlled by the Access Point. To enhance the AP's control capabilities, a new concept of Controlled Access Phase (CAP) has been defined. A CAP is a period of time in which the AP holds control of the channel and stations are not allowed to contend for the medium (Figure 2). An access point can start a CAP by sending a poll or data frame while it finds the medium idle for PIFS waiting time. PIFS is shorter than DIFS or AIFS (used by EDCA), thus giving the AP the capability to interrupt the contention operation and generate a CAP at any moment (after the last frame transmission is complete). Figure 3 depicts this mode of operation. A CFP (as described in 802.11) is also considered a CAP (Figure 2). However, with capability to generate CAPs at any time, there is no need for periodic CFPs. In fact, the 802.11e does not recommend the use of PCF style CFP generation. The CAP generation capability is the main feature that we use for providing per-flow QoS. The 802.11e does not mandate any specific CAP generation discipline and leaves it to system developers to devise such a scheme.

Another new concept that is introduced in 802.11e is Transmission Opportunity (TXOP). A TXOP is a period of time in which a station can hold the medium and transmit multiple frames consequently with SIFS spacing. A station can obtain a TXOP either through contention or through scheduled access, assigned by polling messages. After completion of each frame exchange cycle, if enough time is left in a station's TXOP, it can retain control of the medium and commence a new frame exchange cycle after a SIFS, otherwise it does not continue transmission using SIFS and enters the normal contention mode using AIFS defer and normal backoff.

The 802.11e standard draft provides flow IDs (Traffic Stream ID) in frame formats to enable per-flow QoS handling. It also specifies that it is the responsibility of stations to setup traffic streams (flows) and request resource reservation for them. This is done through sending an ADDTS request to the AP and asking for a traffic stream to be setup with specific traffic specifications. The information carried in the ADDTS request is used by the admission control and scheduling functions of the AP. The ADDTS response by AP completes the traffic stream setup procedure. The standard draft specifies the format in which the traffic stream specifications are described. In fact, we found this description to be very thorough. In particular fields such as service interval and start time are very useful in setting up scheduled access and poll messages.

1.3 Controlled Access Mechanisms

The original 802.11 standard enabled controlled access under the contention free period operation. During a CFP, the Access Point sends poll messages to stations that it assumes have data to send. The CFPs have to be created periodically and the AP could only send poll messages during a CFP. After a CFP ended the AP was not allowed to interrupt the operation and time sensitive data had to contend for the channel. This situation clearly made it difficult to efficiently schedule access to the medium for stations with persistent or periodic data flows. This shortcoming has been addressed in the 802.11e specifications and the AP is allowed to send a poll or data in a contention free manner almost at any time.

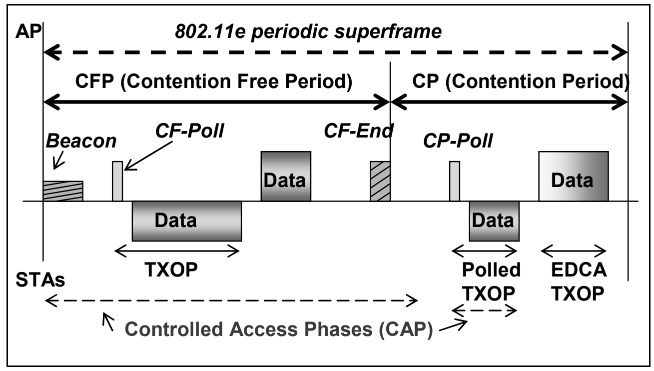

Figure 2. Controlled Access in CFP and CP

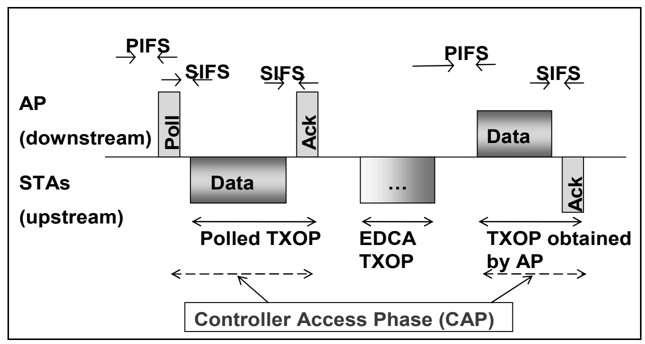

Figure 3 CAP generation during contention period

The mechanisms that are based on the periodic CFP generation usually lack efficiency. We considered providing QoS through periodic CFPs and have devised several algorithms using round robin polling during CFPs [10]. The results, however, showed that the achievable performance using periodic CFPs and round robin methods is limited, mainly due to the inefficiency because of access scheduling only during CFPs as well as the capacity waste due to probing polls that have to be sent (to compensate for the lack of session setup mechanism in 802.11).

The other methods of controlled access can be built around the new features of 802.11e such as CAP and TXOPs. In general a scheduling discipline could be devised to schedule TXOPs for individual stations (including the AP) and assign the TXOP in a CAP. There are many different ways that the scheduling task may be done. However, most conventional scheduling techniques that schedule multiple queues in one node cannot be directly used since the WLAN environment is distributed. These challenges are addressed in the design that we propose in this article and is explained in the following sections.

2. CAPS: Controlled Access Phase Scheduling

The 802.11 WLAN is by design a distributed environment, but the MAC specifications allow the AP to centrally control access to the medium through HCCA features such as CAP generation (Figure 2). We utilized this feature to build a centralized scheduling algorithm that assigns a part of the channel capacity to HCCA operation, essentially for generating CAPs for upstream and downstream flows, which have already reserved bandwidth during session setup. The algorithm allows the remaining capacity to be shared amongst all stations and flows in a contention manner based on EDCA.

To schedule CAPs for both upstream and downstream flows using one central scheduler, we introduce the concept of "virtual packets" (flows). Virtual packets represent, in the AP, the upstream packets of remote stations. This concept enables us to use a single, unified, scheduling framework in the AP that combines the scheduling task for upstream and downstream flows. This framework is called Multiple Access Hybrid Scheduling (MAHS) framework. We first introduced this design as a generic framework in [3]. In this article we describe a new algorithm that customizes this framework for 802.11e WLANs.

2.1 Mixed Downstream/Upstream Scheduling

In 802.11 WLAN, the medium is shared between downstream and upstream traffic in a time-multiplexed manner. Thus, any scheduling discipline must handle packet transmission from individual stations to the AP (i.e. upstream traffic), and packet transmission from AP to the stations (i.e. downstream traffic). Notice that while in the downlink case the packets scheduled for transmission are available in the AP buffers, packets for the uplink flows reside in the stations generating these packets. Since the scheduler resides in the AP, it cannot schedule packets that are waiting in the stations directly. However, the AP can generate a schedule from station requests for packet transmission, and issue polling messages to stations at the times specified by the schedule.

The key to realizing the above scheduling concept, is to represent packets from remote stations (i.e. the upstream packets) by virtual (mini-)packets in the AP, then use a single unified scheduler to schedule virtual packets along with real packets (downstream packets). When scheduling virtual packets, the AP issues polling in the appropriate sequence to generate transmission opportunities for upstream packets. We call this mechanism *hybrid scheduling* because it combines upstream and downstream scheduling in one discipline. The performance of the scheduler will of course depend on the specific scheduling algorithm used. We present a modified version of the well known Weighted Fair Queuing (WFQ, [9]) for this purpose. Using this algorithm ensures bounded delay (thus controlled jitter) and increases the capacity of the system for supporting voice calls.

In our design, depicted in Figure 4, virtual packets are generated by a module called Virtual Packet Generator (VPG). The control plane requests (either direct, through ADDTS message, or implicit through interpreting SIP calls in higher layers, or by traffic pattern estimation) are used by the VPG to determine the patterns of virtual packets (or flows) that must be generated. For example, for a voice call, a periodic flow of packets similar to the real traffic is generated by the VPG in AP. The generated virtual packets are classified along with actual downstream packets and are queued and scheduled for service based on the algorithm which is described in the next section.

Packets that are served by the scheduler are passed to a service classifier. The service classifier is a device that routes actual and virtual packets to different modules. Actual packets are directly transmitted in a downstream CAP, but for virtual packets an upstream CAP is generated by sending a poll message and assigning the appropriate TXOP to the station whose virtual packet is being served.

2.2 Queuing, Scheduling, and Traffic Shaping

The queuing and scheduling model of CAPS is depicted in Figure 4. The integrated scheduler/shaper module combines EDCA and HCCA operation to achieve both fairness and service guarantee. In all stations (including the AP), the queuing model comprises of all queues for flows with reservation (HCCA queues) plus the 8 (or 4) basic EDCA queues. After each

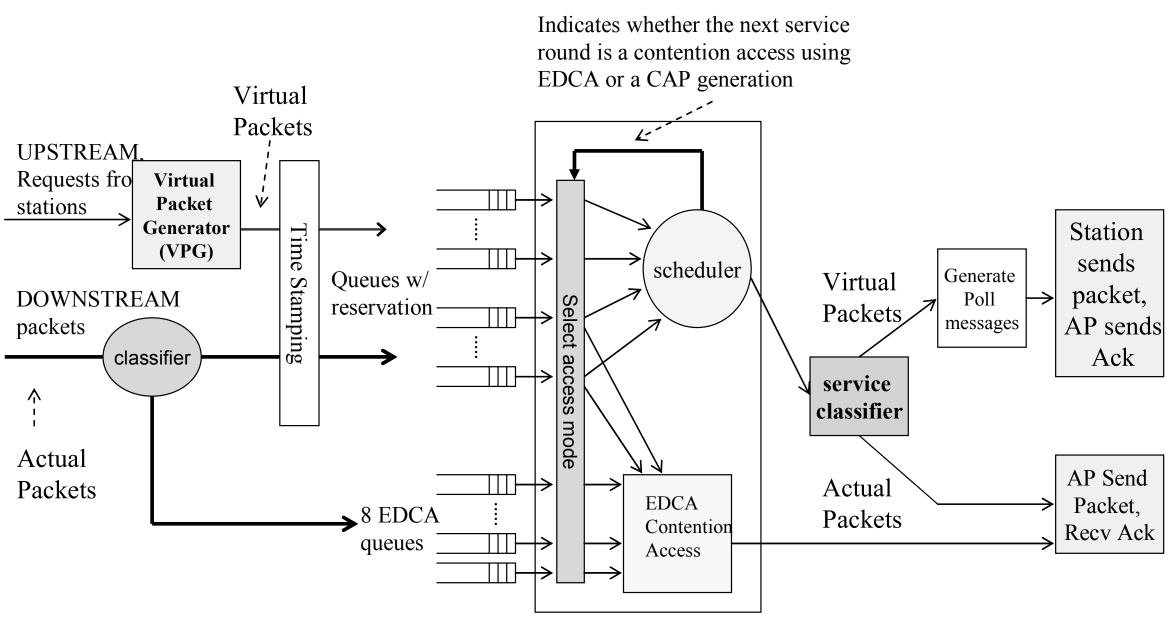

Figure 4 Architecture and queuing model of CAPS in the Access Point

transmission, the scheduler examines the queues with reservation (virtual and actual flow queues) and determines whether a queue must be served. If so, the queue is given controlled access through a CAP generation. But if no queue is chosen for service the scheduler selects the contention access mode and allows all queues in the system to contend for accessing the channel using EDCA rules.

When contention is allowed, all queues in the stations will contend for accessing the channel (including the HCCA queues). But in the AP we only allow EDCA queues plus the actual packet HCCA queues to contend; Virtual flows are excluded from contention because their corresponding actual flows in the stations are already involved in contention. The EDCA contention parameters used by contending HCCA queues are chosen locally based on the information collected during session setup.

The operation of the CAPS algorithm can be divided into three phases, each run in a separate process. The first process is responsible for admission control and generating virtual packets according to declared session information. The second process is responsible for time-stamping and queuing the arriving packets. The third and main process selects the packet to be served and controls the switching between HCCA and EDCA access.

Process 1: Generating Virtual Packets & Admission Control

This process receives requests from stations to set up flows for sessions. Admission control rules are applied to determine whether a session can be admitted by the AP. For an admitted uplink session, this process generates virtual packets using the available information. If service interval S_i and average packet size P_i are specified, virtual flows of size Pi bits are generated every S_i seconds. If S_i is not declared, we can use the declared average rate ri, and generate virtual packets of size P_i every (r_i / P_i) seconds. One way to increase the capacity of the system is to allow bursty transmission through TXOPs and reduce the overhead incurred by poll messages. This is simply achieved by CAPS through using larger virtual packet sizes. For this purpose virtual packet sizes may be increased to match the increase ratio of their service interval. Note that this process provides bandwidth guarantees to flows specified by their average rate requirements. To provide delay guarantees, the maximum burst (b_i) size of each flow i must also be supplied.

Process 2: Queuing Packets

This process receives packets and classifies them into one of three groups 1) virtual packets for actual uplink flows with reservations; 2) real packets belonging to downlink flows with reservations; 3) packets with no flow association and no reservation. The first two types are called HCCA packets in this document, and such packets are assigned to HCCA queues. For scheduling purposes the length attribute of these packets must be adjusted to account for the different overheads incurred by each type. Virtual packets require an extra poll message at the beginning of a CAP, so the transmission period for such packets must be increased to include the poll message time overhead

When a packet without reservation is received, its access category field is examined and the packet is stored in a corresponding EDCA queue. For the HCCA packets, the Traffic Stream ID of the (virtual or real) packet is used to determine its corresponding session queue. Before queuing, HCCA packets are time-stamped according to a WFQ scheduling discipline. In WFQ, the packet finish time is given by the following expression:

$$F_i^k = \frac{L_i^k}{r_i} + \max(F_i^{k-1}, V(t)) \qquad (1)$$

Where F_i^k is the timestamp for the Kth packet from the i^{th} flow, L_i^k is the adjusted packet length, r_i is the rate assigned to the flow, and $V(t)$ is the *virtual time* function. The virtual time, $V(t)$, represents the progress time of a Generalized Processor Sharing scheduler that is being fed with the packets from these queues. More information regarding WFQ can be found in [9] and is not repeated here.

Process 3 Scheduling and Traffic Shaping

This is the main scheduling process and determines the specific action at each service time. A service time occurs after a transmission is completed and the AP has sensed that medium was non-busy for PIFS duration. The algorithm requires maintaining a queue budget parameter g_i for traffic shaping and switching between HCCA and EDCA operations. The queue budget parameter keeps track of the available TXOP time for a specific flow at any given service time, and implements a token bucket traffic shaping scheme. Initially, g_i is set to the maximum allowed burst size of the queue. At each service time the scheduler inspects all HCCA queues to select the Head of Line (HoL) packet with the smallest timestamp and with sufficient g_i (i.e. packet length must be smaller than the queue budget for downlink flows and positive for uplink). If a packet satisfying these conditions is found, then a proportional CAP is generated. Otherwise, control is yielded to EDCA. This algorithm is explained in the following two-step pseudo code format:

```
Step1 // Select the queue to serve:
{   // update budget for all flows.

    Loop ( i : all downlink HCCA queues)
        // t_i :the last visited time for queue i,  t: the current time
        g_i = min( b_i ,  g_i + r_i*(t-t_i) );
        t_i = t;
    End_Loop

    // Find queue i with smallest HoL time stamp
    A1:   find_queue( i :the set of all virtual flow queues plus
                          all downlink HCCA queues with g_i > 0)

    // Apply traffic shaping for Virtual Packets
    if ( i Virtual Packet queue)
    {
        g_i = min { b_i , g_i + vp_size }
        if(g_i < 0)
            discard the HoL virtual packet;
            goto A1;
        else // virtual packet to be served
            goto Step2;
    }
    else if ( i Actual Packet queue)// actual packet to be served
        goto Step2;
    else  // no packet to be served
        goto Step2;
}
```

```
Step2 // Select the operation mode (enforced by mode selector
in Figure 4):
{
    If (no packet is found in Step1)
        exit the algorithm till next round // effectively
                              yielding control to EDCA
    Else
    {
        If (i: Virtual Packet queue)
            send a poll to corresponding station to start a
                              CAP, discard virtual packet
        else if (i: actual packet queue)
            send the packet in a CAP;
    }

    WAIT for response or timeout

    If (    data of size L, sent from HCCA queue i, OR
            received in response to poll from queue i   )
        g_i = g_i - L
    else (timeout or failure)
        do not update g_i

    WAIT until next service round; goto Step1;
}
```

The above algorithm assumes that generated virtual flows are conformant to the reservations made during session setup, but actual downlink or uplink flows may not conform to their previously declared pattern. Therefore, traffic shaping and control is performed differently for actual and virtual flows. The justification is that for uplink flows we only have an estimate of the flow pattern through virtual flow specifications and must wait for the actual packets to be received before we can apply traffic shaping. This fact introduces a one packet lag in the traffic shaping process. For actual flows, we can apply the traffic shaping measures directly to the real downlink flows.

The budget parameter ensures that no flow exceeds its rate reservation. Even if the budget becomes negative due to a large packet being received in response to a small poll, the CAP algorithm holds back the flow until sufficient time passes for its budget to accumulate in the AP. The initial budget is set to the flow's burst size, so that a flow is able to send agreed upon bursts without being penalized. A well behaved flow's virtual packets are never discarded because the AP will generate a poll message for each virtual packet. One additional (optional) adjustment to the algorithm is to allow multiple virtual packets (from one queue) to be sent in one poll message through a larger assigned TXOP, and by increasing g_i for this queue accordingly.

The TXOP that is calculated at each service round is estimated from the budget parameter. For example if a station has budget *g,* and average packets of size *p*, we can assign time for sending approximately $n = g/p$ packets. So the TXOP becomes:

$$txop = min\ \{\ n*(\ (H_m + P + L_{ack})\ /PHY\_Rate + 2*(sifs + H_p/PHY\_Basic\_Rate)\ )\ ,\ MAX\_TXOP\};$$

In the above, H_m is the MAC header, and H_p is the PHY overhead. L_{ack} is the ACK message length. *n* is not rounded up or down to allow a better estimate of TXOP, since stations packets may not all be of size *p*.

3. Performance Evaluation

Using CAPS we achieve better performance in terms of maximum achievable throughput, delay performance and flow protection. In this section we verify these performance gains using simulation experiments. The simulation experiments are conducted using an OPNET based simulator that we have developed for the 802.11e MAC. The simulator accurately simulates the EDCA and HCCA operations and has been verified through comparing the results with those of other simulators such as Network Simulator 2.

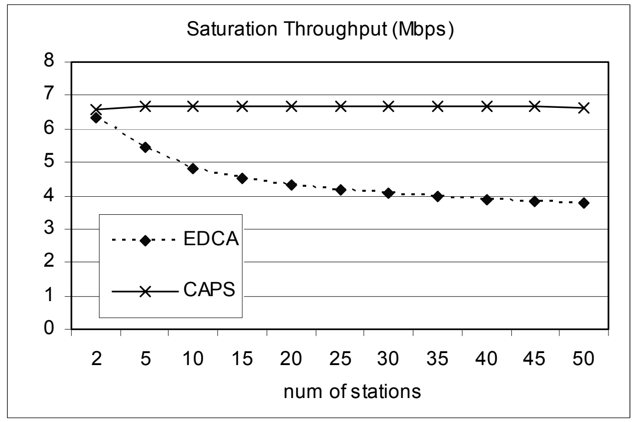

Figure 5 Saturation Throughput

3.1 Maximum Throughput

In the first set of experiments we measured the maximum achievable throughput under saturation conditions for a typical 802.11b network. Under saturation, all stations and queues are always backlogged. The frame sizes in the simulation were uniformly distributed between 50 and 1950 Bytes. As is seen in Figure 5, CAPS outperforms contention based standard EDCA method. It must be noted that although controlled access based mechanisms introduce considerable overhead because of using polling messages, they are still able to provide better services than contention based mechanisms (e.g. EDCA) that perform poorly due to increase in collision. In fact, CAPS algorithm can use the entire channel capacity in a controlled manner and avoid the increase in collision that degrades throughput for EDCA.

Figure 5 shows that CAPS is not sensitive to the increase in the number of stations, while EDCA throughput quickly degrades due to collision. EDCA is very sensitive to the parameters that govern its operation. For example if the default values of the minimum and maximum contention window sizes (e.g. 7 and 15) are used, EDCA throughput quickly drops to almost zero when the number of stations becomes more than 10. For this reason we chose to use large values for contention window sizes (e.g. 256) in the above experiment, to let EDCA achieve better performance; nevertheless, EDCA is outperformed by CAPS. The lower limit on the EDCA saturation throughput is strongly dependent on EDCA parameters and Figure 5 only depicts an example of EDCA's considerable performance degradation and not the absolute values for what one might expect in a typical network (usually lower throughput is observed).

The overhead incurred by the MAC and PHY layers reduces the maximum achievable throughput considerably. Specially, the use of long preambles in PHY intensifies this problem. The use of larger frame sizes or longer TXOPs can alleviate this problem to a great extent. Through using CAPS we can adjust the VP sizes to create larger TXOPs, while maintaining the average rate through spacing the scheduled TXOPs farther from each other (for one flow). In fact, since CAPS provides a deterministic operation environment during saturation (no contention is allowed), we can derive the maximum achievable throughput as follows:

$$n = [\frac{TXOP}{2*SIFS + (P + H_m + L_{ack})/PHY\_Rate + 2*H_p}] \quad (2)$$

$$T = (2n+1).H_p + \frac{L_{poll} + n.(H_m + P + L_{ack})}{PHY\_Rate} + PIFS + 2n.SIFS \quad (3)$$

*throughput = (n*P/T)* (4)

In the above, n is the number of frames of size *P* that can be transmitted in one *TXOP* duration. *T* is the actual time that it takes to send *n* packets of size *P* (therefore *T* is less than *TXOP*). L_{poll} is the Poll message length. We calculated the achievable throughput for several TXOP sizes (packet size fixed at 1000 Bytes). The measurements from simulation experiments were also collected. Figure 6 depicts the results. As is seen, at around 8 milliseconds, the throughput does not increase as much if TXOP is increased. Interestingly the maximum TXOP limit specified in the standard is 8.192 milliseconds which provides a good balance between achieving higher throughputs and limiting TXOP.

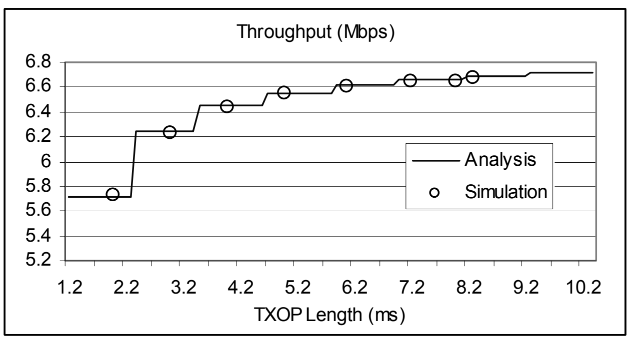

Figure 6 Maximum Throughput vs. Frame Length

3.2 Delay Performance

In the second set of experiments we studied the delay performance of the CAPS algorithm. These experiments were done for different types of multimedia traffic. First, we used 64 Kbps voice streams and observed that CAPS is able to accommodate considerably higher number of voice sessions compare to the standard EDCA operation. In this experiment no background or data traffic was present. As is shown in Figure 7, the average and especially the maximum delay for voice sessions remain controlled for a higher number of voice sessions when CAPS is used. This is expected as CAPS tries to assign guaranteed and consistent service to each flow.

For the next experiment we observed the average delay for a 256Kbps variable bitrate H.264 video flow while the background data traffic was being increased. The results, depicted in Figure 8, show that CAPS is able to protect the video flow from the increasing background traffic, even when the network enters saturation operation. For EDCA, the delay increases quickly as the offered load increases and nears the maximum capacity.

In these experiments we increased the default minimum and maximum contention window sizes for EDCA voice and video access categories to let it accommodate more stations. Without this increase EDCA will fail very quickly as the number of stations increases. We also allowed larger virtual packets, but with longer service intervals to allow for bursty operation in CAPS (EDCA by default uses bursty operation for voice and video categories).

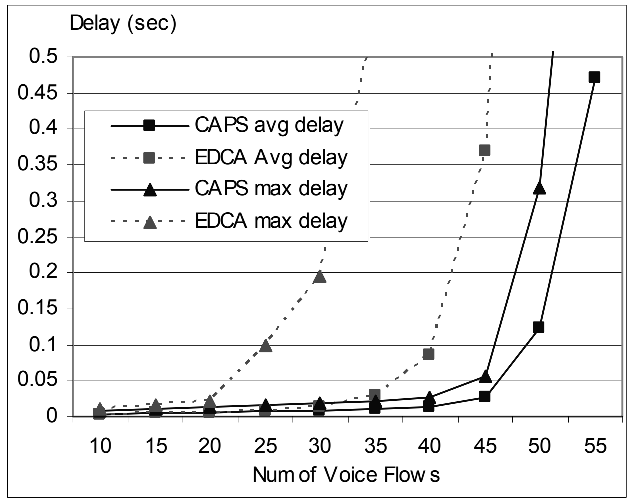

Figure 7 Average delay for a Voice only WLAN

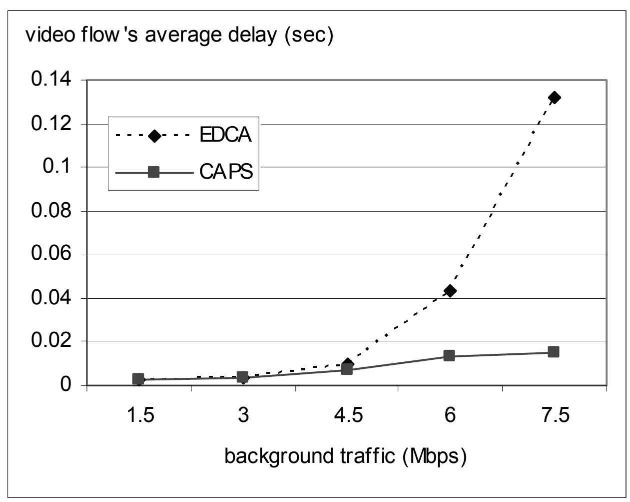

Figure 8 Average Delay for a Video flow as background traffic increases

To further observe the ability of CAPS to protect flows from other traffic in the channel and its guaranteed service provisioning feature, we conducted an experiment to observe the behaviour of CAPS and EDCA in a scenario in which a low bitrate multimedia flow had to compete with the same priority high bitrate flows. The low bitrate flow in this experiment was a 66 Kbps H.264 stream, while the higher bitrate flows were a number of 256 Kbps streams. We increased the number of 256 Kbps flows and observed the delay performance. All flows in CAPS had their respective VP's generated at the same rate of the traffic and the WFQ scheduler's weights were set according to the average rate of each flow.

Since EDCA is inherently an aggregate service differentiation protocol we expect it to fail to isolate flows within the same priority class. This means that EDCA will fail to protect the low bitrate flow from the same class traffic. This is indeed what we observed. On the other hand, CAPS managed to efficiently protect the observed low bitrate flow. The measurements from this experiment are depicted in Figure 9. The background data traffic in this case was 4Mbps.

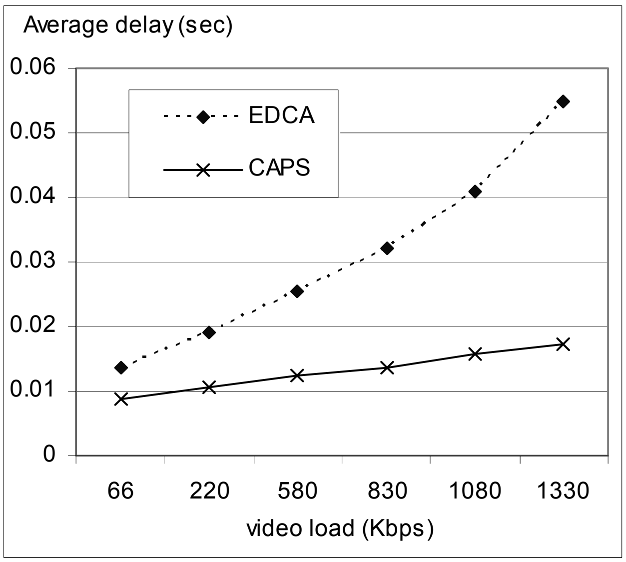

Figure 9 Average delay of a low bitrate flow in presence of same class traffic

4. CONCLUDING REMARKS

Supporting multimedia applications is an important requirement for present and future WLAN technologies. The IEEE 802.11 standard, which is the most popular WLAN technology today, has not been originally developed to support demanding multimedia applications such as voice and video conferencing. To address this issue, an enhanced version of this standard is being developed under the IEEE 802.11e standard. Although, 802.11e provides enough features and capabilities to provide QoS, it does not provide a final solution and leaves it to vendors and implementers to use the available features and devise their own QoS solution.

We use the new capabilities, offered by 802.11e, and design an algorithm and framework that is able to provide per flow service guarantees (given ideal channel conditions) in the MAC layer of an 802.11e WLAN. In particular, our solution is based on the HCCA features of the upcoming 802.11e standard. The proposed design enables centralized scheduling of upstream and downstream flows in the access point. It also facilitates on demand use of controlled access phases under HCCA, while allowing EDCA operation for the remaining capacity. This feature allows very efficient service guarantee for time sensitive flows even under heavy traffic conditions. For example, multimedia sessions such as voice and video can be given guaranteed access to the medium through HCCA controlled access phases, while the remaining capacity is used by the background traffic in a contention manner. We have conducted several experiments with different operation scenarios and demonstrated the superior performance of CAPS compare to EDCA.

Currently we are examining the effects of multi-rate operation of the PHY layer on the efficiency of our design and will consider enhancing CAPS in order to efficiently address the changes in channel quality. Other open research issues that arise from this work are the application of our design to other networks such as IEEE 802.16, and the integration of the presented design with the power management features of 802.11e.

5. REFERENCES

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.

[2] IEEE Draft Std. 802.11e/D8.0, "Medium Access Control (MAC) Quality of Service (QoS) Enhancements," Feb. 2004.

[3] Pourmohammadi-Fallah Y., Elfeitori, A., Alnuweiri. H. , "A Unified Scheduling Approach for Guaranteed Services over IEEE 802.11e Wireless LANs", Broadband Networks, 2004. First Int. Conf. on, Oct. 2004 Page(s):375 – 384

[4] J. Rosenberg, et.al., "SIP: Session Initiation Protocol", RFC 3261, June 2002

[5] Qiang Ni, Lamia Romdhani, and Thierry Turletti. "A Survey of QoS Enhancements for IEEE 802.11 Wireless LAN". Wiley Journal of Wireless Communication and Mobile Computing (JWCMC), John Wiley and Sons Ltd., 2004; Volume 4, Issue 5: 547-566.

[6] Ranasinghe, R.S.; Andrew, L.L.H.; Hayes, D.A.; Everitt, D.;" Scheduling disciplines for multimedia WLANs: embedded round robin and wireless dual queue", Communications, 2001. ICC 2001. IEEE International Conf. on , Volume: 4 , 11-14 June 2001 pp:1243 - 1248

[7] Pattara-Atikom, W.; Krishnamurthy, P.; Banerjee, S.;" Distributed mechanisms for quality of service in wireless LANs", Wireless Communications, IEEE, Volume: 10 , Issue: 3 , June 2003 Pages:26 – 34

[8] Banchs, A.; Radimirsch, M.; Perez, X.;" Assured and expedited forwarding extensions for IEEE 802.11 wireless LAN" Quality of Service, 2002. Tenth IEEE International Workshop on , 15-17 May 2002 Pages:237 - 246

[9] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," IEEE/ACM Trans. on Networking, vol.1, no. 3, pp. 344-357, June 93.

[10] Pourmohammadi Fallah Y., Alnuweiri H., "Enhanced Controlled-Access and Contention-Based Algorithms for IEEE 802.11e Wireless LAN",to appear in proceeding of Wirelesscom 2005, Int. Conf. on Wireless Networks, Communications, and Mobile Computing, June 2005